REFINEMENTS IN ITERATIVE BLIND DECONVOLUTION

A thesis presented for the degree of
Doctor of Philosophy in Electrical and Electronic Engineering
at the University of Canterbury, Christchurch, New Zealand.

July 1995

Hong Jiang

ABSTRACT

The thesis presents a number of results regarding blind deconvolution of a single contaminated blurred image.

A scheme of routinely selecting the best support is proposed. This scheme employs an objective error metric to evaluate the performance of Davey's blind deconvolution algorithm [Davey et al, Optics Comms., 69:356, 1989]. The effectiveness of this scheme for choosing the best support sizes as determined by testing is demonstrated.

Two new iterative algorithms are presented for solving blind deconvolution problems. Both algorithms incorporate a number of refinements into the Davey Algorithm. The performance of the first algorithm, referred to as the Automatic Iterative Algorithm (AIA), has been extensively examined in deconvolving images contaminated with different levels of Gaussian noise. The superiority of the AIA over the Davey Algorithm is demonstrated in terms of ease of parameter selection, computational efficiency and accuracy of reconstructions. The second algorithm referred to as the Coloured Noise Algorithm (CNA) is designed for deconvolving images contaminated with coloured noise. The superior performance of the algorithm over the Davey Algorithm in the presence of such noise is also demonstrated.

Various existing blind deconvolution techniques are reviewed. A comparison of the success of two techniques in solving the same problem is presented. The first new algorithm referred to above is chosen to represent iterative methods and is demonstrated to converge faster and produce better quality restorations for contaminated images than a representative optimization method (conjugate gradient).

Finally, computational simulations related to the contention that redundancy exists in multidimensional blind deconvolution are presented. The study suggests that blind deconvolution in three or more dimensions is overdetermined by sampling the convolution at the Nyquist rate.

ACKNOWLEDGEMENTS

This thesis would not come true without the support and encouragement of many people. I am eternally grateful to my original supervisor, the late Professor Richard Bates, for his guidance and encouragement. His enthusiasm for new ideas and his pursuit of excellence has been a great inspiration to me.

I am greatly indebted to my current supervisor, Dr Phil Bones, for all his effort to enable me to complete this study. His questioning, suggestions, encouragement and many hours he has spent on proof reading this document have significantly improved its quality.

I am particularly grateful to Bruce McCallum for his insight, input and stimulating discussions. I am grateful to Bruce Davey for his help to introduce me to the subject. I am thankful to Brenda Satherley and John Kirkland for their help and effort on proof reading this document. I thank many staff members and students both in and outside of the Department for their help and friendship, especially Charles Parker, Russell Watson, Richard Lane, Bonnie Law, Pat Roddick, Rick Millane, Richard Fright, Peter Gough, Mike Shurety, Dave Van Leeuwen, Nigel Brieseman, Donna Ward and Felix Hartanto. Special thanks to Yihong Hu for her companionship, many interesting discussions and help when we both spent long hours in writing theses.

I greatly appreciate the love, support and patience of my husband Zhengzhe. His help and understanding during the final stage of writing this document have been invaluable. I deeply appreciate the enduring love of my parents. This thesis is a product of their examples, support and sacrifices. I am grateful to my sister Qin for her encouragement, understanding and advice.

I am grateful to many friends and relatives who have generously forgiven me for not writing to them as often as I should have due to this thesis. I thank those friends who have made my stay in Christchurch enjoyable. In particular, I would like to thank the past and present members of R6, Sandie, Martin, Bob, Jane and Briar. My sincere thanks go to Winton and Jean for their friendship and preparing and delivering many delicious weekend meals to me during the last few months.

I gratefully acknowledge the financial support from the New Zealand Ministry of

External Relations and Trade and the Royal Society of New Zealand - Canterbury Branch Travel fund.

CONTENTS

PREFACE

This preface describes the background of the study covered in this thesis.

In 1988, I had the privilege to start my postgraduate study under the late Professor Richard H. T. Bates, who was well known for his contribution and active involvement in multidisciplinary research areas, including image processing, speech processing, inverse problem, radio antenna engineering.

From my start as a Master's student and later as a PhD student, I have been involved in the field of image processing. I joined another postgraduate Bruce Davey (Davey, 1989) in study of the blind deconvolution problem. At that time the blind deconvolution problem had just been analytically proven to be potentially solvable (Lane and Bates, 1987a) and the earliest practical blind deconvolution algorithms were coming out (Lane and Bates, 1987b; Ayers and Dainty, 1988; Davey *et al.*, 1989), so a lot of attention was drawn in this newly developing area in Professor Bates' image processing group. Initially I worked on both of recovering astronomical images using the shift-and-add algorithms (Bates, 1976; Bates and Cady, 1980; Bates and Davey, 1987) and solving the pure blind deconvolution problem employing two algorithms devised by Lane and Bates (1987b) and Davey *et al.* (1989). As a result of this study, I assisted Professor Bates to prepare a chapter of a book on the subject of blind deconvolution (Bates and Jiang, 1991). During the period I also coauthored (with Professor Bates) to prepare a paper on iterative blind deconvolution; an idea (in concept only) of routine implementation of blind deconvolution, in terms of controlling the filter constant and the support estimate, was suggested in this paper, and this directed my further study as reported in Chapters 4 and 5. It was fortunate that at that time Bruce Davey and Bruce McCallum (McCallum, 1990b) were in the Department in preparing their PhD theses and both had substantial experience in the area. Sadly, Professor Bates passed away in November 1990. It was fortunate that Dr Phil Bones became my supervisor afterwards.

By the time I met Bruce McCallum (when he visited the Department in February 1993), we both had devised different schemes for routinely controlling the filter constant in iterative blind deconvolution. Noticing that the results using a larger initial filter constant with my strategy was more stable than using a smaller one, we saw the potential

advantage of using the strategy which led to the study and the methods described in §5.2.2.1 to §5.2.2.3.

Also in February 1993 Professor Rick Millane (from Purdue University, Indiana) was visiting the Department. Professor Millane had proposed a theory on redundancy in multidimensional blind deconvolution and was keen on testing its implications for solving blind deconvolution problems using simulated data. I was also interested in the subject. Together with Dr Bones, we started the work. Results were reported in Millane *et al.* (1994) and chapter 7.

This thesis contains 8 chapters. The outline of each chapter is now described.

Chapter 1 contains a brief overview of imaging science and degradations which occur in measuring images. Various types of deconvolution problems are introduced and the one this thesis focuses on is pointed out. This chapter concludes with a brief introduction of the original work presented in this thesis.

Chapter 2 introduces the notation, terminology and mathematical tools used in the thesis.

Chapter 3 reviews various techniques available for solving pure blind deconvolution problems, beginning with the two most relevant standard deconvolution techniques.

Chapter 4 presents original investigation and analysis of important aspects of the Davey Algorithm (the computational simulation contained in §4.1.2 was initially investigated by Davey (1989)), including: the effect of the key parameters on its convergence; derivation of the least squares filter for blind deconvolution; the roles of the Wiener-like filter, its filter constant and exponent in the algorithm's performance; discussion on a suggestion made by Seldin and Fienup (1990); and the effect of the objective error measures on monitoring the performance of the algorithm. Finally, a method of routinely selecting the best support is proposed. The above study supplies a sound experimental and analytical basis for the formation of the new algorithms presented in chapter 5.

Chapter 5 presents two new blind deconvolution algorithms which are both based on the Davey Algorithm. The first algorithm, described in §5.2, includes various refinements to the Davey Algorithm, such as routinely controlling the filter constant and interpolation of the spectral estimates of the image and the psf. The performance of the algorithm, in terms of reducing computational cost, accelerating its convergence and improving the accuracy of the restorations, is demonstrated.

The second algorithm described in §5.3 is developed to handle images contaminated by coloured noise.

Chapter 6 presents a comparison of the effectiveness of two blind deconvolution algorithms on solving the same problem. These two algorithms belong to two different classes, namely the iterative methods and optimization methods.

Chapter 7 presents original computational simulations related to the contention of redundancy in multidimensional blind deconvolution. Both 2 and 3 dimensional problems are studied.

In the final chapter, conclusions on the original work covered in chapters 4 to 7 are drawn and suggestions for further research are discussed.

The software employed to produce original results contained in this thesis is the image processing package *improc*, mainly developed by R. G. Lane and B. L. K. Davey. The routines used to perform my algorithms were written by myself. The package and the conjugate gradient blind deconvolution algorithm used to generate results of the optimization method in chapter 5 was written by R. G. Lane and B. F. Law.

Publications and presentations made during the cause of this study are listed in the following:

BATES, R.H.T., JIANG, H. and DAVEY, B.L.K. (1990), 'Multidimensional system identification through blind deconvolution', *Multidimensional Systems and Signal Processing*, Vol. 1, pp. 127–142.

JIANG, H. and BONES, P.J. (1991), 'Estimation of best support size in iterative blind deconvolution', In *Proceedings 6th New Zealand Image Processing Workshop*, Lower Hutt, August, pp. 127–132.

BATES, R.H.T. and JIANG, H. (1991), 'Blind Deconvolution – Recovering the Seemingly Irrecoverable!', In GOODMAN, J.W. (Ed.) *International Trends In Optics*, Academic Press, San Diego.

BONES, P.J., WATSON, R.W., JIANG, H. and SATHERLEY, B.L. (1992), 'Restoring images - after the Bates tradition', In *Proceedings IPENZ Annual Conference*, Christchurch, February, Vol. 2, pp. 165–173.

MILLANE, R.P., BONES, P.J. and JIANG, H. (1994), 'Blind deconvolution for multidimensional images', In *IEEE International Conference ASSP*, Adelaide, Australia, April, Vol. 5, pp. 445–448.

GLOSSARY

The symbols and abbreviations used in this thesis are defined in the following:

Symbols

Chapter 1

$\odot$	the convolution operator

Chapter 2

$\|q(\mathbf{x})\|$	the magnitude of $q(\mathbf{x})$
$\mathcal{P}[q(\mathbf{x})]$	the phase of $q(\mathbf{x})$
$Re[q(\mathbf{x})]$	the real part of $q(\mathbf{x})$
$Im[q(\mathbf{x})]$	the imaginary part of $q(\mathbf{x})$
$q^*(\mathbf{x})$	the complex conjugate of $q(\mathbf{x})$
SNR	the signal to noise ratio of an image
Eng$[q(\mathbf{x})]$	the energy of $q(\mathbf{x})$
$S_q(\mathbf{x})$	the support of $q(\mathbf{x})$
$B_q(\mathbf{x})$	the image box of $q(\mathbf{x})$
$L_q(x)$	the extent of $q(\mathbf{x})$ in the x direction
$\triangle$	the sampling interval
$\delta(\mathbf{x})$	the delta function or impulse
$*$	the correlation operator

Chapter 3

$I(\mathbf{u})$	the inverse filter
$W(\mathbf{u})$	the standard Wiener filter
$\in$	is an element of
$\notin$	is not an element of

$q_i(\mathbf{x})$	the estimate of $q(\mathbf{x})$ at i^{th} iteration before image space constraints
$\hat{q}_i(\mathbf{x})$	the estimate of $q(\mathbf{x})$ at i^{th} iteration after image space constraints
$W_{\hat{Q}_i}(\mathbf{u})$	the Wiener-like filter formed by $\hat{Q}_i(\mathbf{u})$
Ω_i^q	the set of pixels at which $q_i(\mathbf{x})$ violate image space constraints
$E_I[q_i(\mathbf{x})]$	the image space error of $q_i(\mathbf{x})$
$E_T[q_i(\mathbf{x})]$	the true error of $q_i(\mathbf{x})$
$Z[Q_i(\mathbf{u})]$	zero-sheet of $Q_i(\mathbf{u})$

Chapter 4

$M_q(\mathbf{x})$	the estimate of the support of $q(\mathbf{x})$
Q_{max}	the cut-off spectral magnitude of $\hat{Q}_i(\mathbf{u})$

Abbreviations

FFT	fast Fourier transform
IFFT	inverse fast Fourier transform
DFT	discrete Fourier transform
AIA	automatic iterative algorithm
CNA	coloured noise algorithm
CGA	conjugate gradient algorithm
psf	point spread function
RHS	right hand side

Chapter 1

INTRODUCTION

This thesis is concerned with the problem of image restoration which occurs in diverse areas in science and engineering. The nature of images and the ways in which they may be degraded are introduced in §1.1. A common mathematical model of the degradation, a convolutional model which is the basis of work presented in this thesis, is described in §1.2. One particular image restoration problem, blind deconvolution, is introduced in §1.3, wherein the main contribution of this thesis is introduced.

1.1 IMAGES AND DEGRADATION

In our fast growing world, knowledge of many diverse physical entities (objects) are demanded to satisfy human beings' ongoing quest for understanding and exploring this world. Information about numerous of these entities are collected as images by the use of imaging instruments. Celestial objects are captured by optical or radio astronomical telescopes. Maps of pollution density and distribution can be obtained by satellite imaging. Information about organs within the human body can be recorded by medical imaging, for example by X-rays computed tomography and magnetic resonance imaging. A map of underground structure can be derived from seismic data. Many more measurements can be found in oceanic research, space exploration, microscopy, weather prediction, crop assessment, biology, nuclear medicine, law enforcement, defense and industrial applications.

The quality of the measurement is critical if an image is to accurately represent the object. The resolution of any measurement is limited and measurements are inevitably corrupted by noise, such as film-grain noise or thermal noise when a electronic imaging sensor is used. Measurements can be degraded by imprecision of the imaging instrument, for example by diffraction and various aberrations of the lenses in an optical imaging instrument. Spherical aberration seriously degrades the image of the Hubble Space Telescope due to the large size of its primary lens (Katsaggelos, 1991). Measurements can also be degraded by imperfections in the imaging environment, such as vibration of the imaging system or atmospheric turbulence in remote sensing and aerial imaging.

1.2 DEGRADATION MODEL

To effectively restore an image from degraded measurements, an accurate mathematical model describing the effects of the image degradation from the imaging process is critical. The model assumes the existence of an ideal or true image, that contains undistorted information of the object on which the degraded image is based.

A general degradation model describing the relationship of the true image, denoted $f(\mathbf{x})$, and the contaminated blurred image, defined $g(\mathbf{x})$, is

$$g(\mathbf{x}) = S\left[\int_{\mathbf{x}'} f(\mathbf{x}')h(\mathbf{x},\mathbf{x}')\,d\mathbf{x}'\right] + c(\mathbf{x}), \tag{1.1}$$

where $S[\cdot]$ represents a function which may be nonlinear, $h(\mathbf{x},\mathbf{x}')$ is the system response function to a point in $f(\mathbf{x})$ at location $\mathbf{x}'$; $c(\mathbf{x})$ represents *contamination* of the blurred image $g(\mathbf{x})$; $\mathbf{x}$ and $\mathbf{x}'$ are the position vectors of arbitrary points in the K-dimensional space, which is called the *image space*. In majority of cases considered in this thesis, $K = 2$; while $K = 3$ in situations discussed in chapter 7.

Consider now the effect of each type of degradation on the true image $f(\mathbf{x})$. Degradation caused by nonlinear response of the imaging system usually occurs in the recording phase. When a photographic image is involved, the density of the silver halide grains deposited on the developed paper is proportional to the logarithm of the light intensity that exposed the film (Pratt, 1991). When a photoelectrical sensor is used, for example the charged coupled device (CCD), the output current of the sensor is in proportion to the logarithm of the incident light. However, the nonlinear response of the recording devices can be compensated for and prevented if the nonlinear characteristic of the film or sensor is known. Under this assumption, the process of the image formation can be considered as a linear process. Equation (1.1) can thus be approximated as

$$g(\mathbf{x}) = \int_{\mathbf{x}'} f(\mathbf{x}')h(\mathbf{x},\mathbf{x}')\,d\mathbf{x}' + c(\mathbf{x}). \tag{1.2}$$

The system response function $h(\mathbf{x},\mathbf{x}')$ in (1.2), which is also named the *blurring function* or *point spread function* (psf), represents the second type of degradation which degrades the true image. This degradation is introduced by an imaging system operated under a nonideal condition. Two common examples of blurring in optical imaging are motion blur and out-of-focus blur. The former occurs when relative motion exists between the object and the imaging instrument. The blurring introduced by the imperfectness of the imaging lenses is also under this category. Equation (1.2) implies that the response of the imaging system to a point source is dependent on the location of the point in the image space. That is the blurring function is *space variant*. There are situations where the blurring function can be considered *space invariant* or *isoplanatic*. The generalized blurring pattern of $h(\mathbf{x},\mathbf{x}')$ can thus be reduced to the form $h(\mathbf{x}-\mathbf{x}')$ from which it can be seen to be independent of location on the true image. Consequently,

in the absence of contamination, the relationship of the output blurred image and the input true image in (1.2) becomes a convolution

$$b(\mathbf{x}) = \int_{\mathbf{x}'} f(\mathbf{x}')h(\mathbf{x} - \mathbf{x}')\,\mathrm{d}\mathbf{x}', \tag{1.3}$$

which may be written as

$$b(\mathbf{x}) = f(\mathbf{x}) \odot h(\mathbf{x}), \tag{1.4}$$

where $\odot$ represents the convolution operator in K-dimensions. In the presence of contamination, the model for the blurred image is

$$g(\mathbf{x}) = f(\mathbf{x}) \odot h(\mathbf{x}) + c(\mathbf{x}). \tag{1.5}$$

The degradation caused by the contamination term $c(\mathbf{x})$ in (1.5) mainly originates from the recording system of the imaging instrument. Examples are film-grain noise due to film-grain irregularities and thermal noise resulting from random electron fluctuations in resistive elements of photodetectors or resistors within sensor amplifiers. These two types of noise can be approximated by an additive Gaussian process and zero-mean additive Gaussian process respectively (Andrews and Hunt, 1977), where the term additive implies that the noise is uncorrelated with the true image. A small deviation of the blurred image from the convolutional model, such as may be caused by slight nonlinearity of the recording system and/or nonisoplanatism of the blurring function, can be incorporated into $c(\mathbf{x})$.

The degradation model in (1.5) is an approximation of the general model in (1.1). It has been widely used in image restoration research and is the basis of the methods presented herein.

1.3 BLIND DECONVOLUTION PROBLEMS

Image restoration problems can be classified into different categories according to the type of measurements and prior knowledge about the blurring available. Since the blurred image may be represented by a contaminated convolution, the restoration to obtain an image estimate is commonly termed *deconvolution.*

The most common deconvolution problem is *standard deconvolution* problem which is posed as:

> "Given a blurred and contaminated image $g(\mathbf{x})$ modelled by equation (1.5) and an estimate of psf $h(\mathbf{x})$, recover an estimate of the image $f(\mathbf{x})$."

Inversion of (1.5) to estimate $f(\mathbf{x})$ from given measurement $g(\mathbf{x})$ and blurring function $h(\mathbf{x})$ is often not as simple as it appears. The deconvolution operation is very sensitive to contamination. Small errors involved in the measurement can be exaggerated in the

restoration of the image (Riad, 1986). During last 30 years, many techniques have been developed for solving standard deconvolution problems. Some of these techniques along with the limitations they suffer are reviewed and discussed in §3.1.

In many situations the blurring function of the imaging system is unknown. The deconvolution then becomes *blind*, and a much more difficult task to solve (Bates and Jiang, 1991). Blind deconvolution was introduced originally to denote image recovery from a large collection of differently blurred versions of an image (Labeyrie, 1970; Stockham *et al.*, 1975). This type of deconvolution is referred as *ensemble blind deconvolution*, posed as:

"Recover an estimate of $f(\mathbf{x})$ from a given ensemble $\{g_m(\mathbf{x}); m = 1, 2, ..., M\}$ of independently blurred and contaminated versions of $f(\mathbf{x})$".

The ensemble blurred images is described as

$$g_m(\mathbf{x}) = f(\mathbf{x}) \odot h_m(\mathbf{x}) + c_m(\mathbf{x}), \tag{1.6}$$

where $h_m(\mathbf{x})$ and $c_m(\mathbf{x})$ are statistically independent members of the ensembles of psfs and contaminations respectively, while the true image, which is the basis of each member of the ensemble, is identical. Ensemble blind deconvolution problems occur, for example, in astronomical imaging (Bates, 1982; Roddier, 1988), ultrasonic imaging (Bates and Minard, 1984), restoring audio recordings (Stockham *et al.*, 1975) and speech processing (Thorpe, 1990, §4.2). Several methods have been developed to attack these problems over the last 20 years.

When only a single blurred image is known, the deconvolution is termed the *pure blind deconvolution* problem, which is posed as:

"Given a single blurred image $g(\mathbf{x})$ only, recover an estimate of the image $f(\mathbf{x})$."

The pure blind deconvolution presents a more general and most difficult problem since the least information is available. In the absence of contamination, however, a solution to the blind deconvolution problem exists and is unique if both $f(\mathbf{x})$ and $h(\mathbf{x})$ are compact (*cf.* §2.2) and irreducible (*cf.* §2.4.2.1) and both have more than one dimension. This conclusion derives from the concept of zero-sheets by Lane and Bates (1987a) (*cf.* Lane *et al.*, 1987). Within the last decade, pure blind deconvolution has become practical to solve (Lane and Bates, 1987b; Ayers and Dainty, 1988; Davey *et al.*, 1989) and has attracted a lot more attention recently (McCallum, 1990b; Lane, 1992; Holmes, 1992) . This thesis focuses on aspects of iterative solution to this problem. Various pure blind deconvolution techniques are reviewed in chapter 3. For simplicity, the term blind deconvolution is used to mean pure blind deconvolution in the rest of the thesis unless otherwise stated.

A special case of the blind deconvolution problem is the *phase retrieval* problem, which can be stated:

"Given the Fourier magnitude of $f(\mathbf{x})$, $|F(\mathbf{u})|$, recover an estimate of $f(\mathbf{x})$."

The inverse Fourier transform of the power spectrum (magnitude squared) of $f(\mathbf{x})$, is the autocorrelation of $f(\mathbf{x})$ (*cf.* §2.4.2.2). Replacing the psf $h(\mathbf{x})$ in (1.5) by $f^*(-\mathbf{x})$, the blurred image $g(\mathbf{x})$ can be seen to become a contaminated version of the autocorrelation of $f(\mathbf{x})$. Blind deconvolution thus reduces to retrieving the Fourier phase of $f(\mathbf{x})$ from the given $|F(\mathbf{u})|$. Phase retrieval problems have been the subject of research for longer than blind deconvolution and several well-known algorithms are in use (Fienup, 1982). Applications of phase retrieval methods involve the fields such as electron microscopy (Gerchberg and Saxton, 1972; Misell, 1978), X-ray crystallography (Ramachandran and Srinivasan, 1970; Millane, 1990a), and astronomical imaging (Bates, 1982).

The original work presented in this thesis is in two parts. The first and the main part of the research is focused on developing new blind deconvolution algorithms. This work started with an extensive study and further investigation of one of the most effective blind deconvolution algorithms as presented in chapter 4, where the rationale of the central part of the algorithm is analyzed with examples, questions posed by other authors are discussed and a method for estimating a key parameter for blind deconvolution, the support, is presented. Based on this study, a much more efficient and effective algorithm, the Automatic Iterative Algorithm (AIA), which is iterative in nature, is proposed in §5.2. The new algorithm AIA converges faster and more accurately than its antecedents. The AIA seems to be capable of retrieving some information which is beyond a practical band-limit imposed by contamination (within this bandwidth the magnitude of the spectrum of an image is considered to be no less than the level of contamination) of spectrum of $g(\mathbf{x})$. Unlike most of other iterative algorithms, the progress of the AIA towards convergence can be monitored by means of a reliable (objective) error metric. The objective error metric correlates well with the true error metric in general, which makes it possible to record satisfactory restorations without the reference of the true image (which is not available in practice). A second new algorithm, the Coloured Noise Algorithm (CNA), is developed for deconvolving images which are contaminated by coloured noise (*cf.* §5.3).

While several blind deconvolution algorithms have been reported here and elsewhere, information about the comparative effectiveness of these algorithms in recovering images is, however, little reported. In chapter 6 two representative algorithms that belong to different classes are chosen for such a comparison. One is the first new algorithm, the AIA, presented in this thesis (*cf.* §5.2). The other algorithm standing for the optimization methods is conjugate gradient algorithm (*cf.* §3.4.2). All comparative results show that the convergence of the former algorithm is significantly faster than the latter in terms of the number of iterations and the absolute time required, and the results also demonstrate that the former algorithm produces better quality restorations in the presence of contamination than the latter.

The second part of the original work involved in the thesis is a demonstration Millane's contention that the three (or more) dimensional image is overdetermined by its Nyquist samples (Millane, 1990b). A computational simulation of blind deconvolution to demonstrate this theory is presented in chapter 7. An implication of Millane's work and that presented here is that solving blind deconvolution problems in three or higher dimensions may be easier than in two-dimensions.

Chapter 2

PRELIMINARIES

The purpose of this chapter is to introduce the notation, terminology and mathematical theory which are used in the subsequent chapters. Mathematical notation used to describe quantities such as vectors, functions, complex numbers is included in §2.1. Two important deconvolution concepts of compact image and image form are introduced in §2.2 and §2.3. The image formation process, modelled as an approximation of the linear and position invariant process §1.2, makes Fourier theory to be an extremely useful and powerful mathematical tool for image restoration. Useful Fourier theory and properties are described in §2.4.

2.1 NOTATION

Throughout this thesis a K-dimensional vector is indicated by a bold symbol, e.g. $\mathbf{x}$, and the k^{th} component of the vector $\mathbf{x}$ is represented by $\mathbf{x}$ with a subscript of k, i.e. $\mathbf{x}_k$. Images are frequently Fourier transformed into *Fourier space*. The domain where an image or an estimate of the object exists is called the *image space* (or the spacial domain). A lower case symbol represents a quantity in the image space (an image) and the corresponding upper case letter denotes the Fourier transform of the image. The variables $\mathbf{x}$ and $\mathbf{u}$ indicate arbitrary position vectors of the Cartesian coordinate system in K-dimensional image space and Fourier space respectively. Thus, a two-dimensional image is represented by $f(x, y)$ or $f(\mathbf{x})$ and its Fourier transform is denoted $F(u, v)$ or $F(\mathbf{u})$. When the variable itself is a function, it is described by a set of square brackets. For example, $S[f(\mathbf{x})]$ represets a function S of the variable $f(\mathbf{x})$. In this thesis, the majority of the images concerned are two-dimensional.

The real and imaginary parts of a complex quantity c are described by $Re(c)$ and $Im(c)$ respectively. Thus

$$c = Re(c) + iIm(c), \tag{2.1}$$

where $i = \sqrt{-1}$. Alternatively c may be expressed in polar coordinates as

$$c = |c| \exp^{\mathcal{P}(c)}. \tag{2.2}$$

The signal-to-noise ratio of an image with additive contamination, for example $g(\mathbf{x})$ in (1.5), is defined as

$$SNR = 10 \log \left(\frac{\int |f(\mathbf{x}) \odot h(\mathbf{x})|^2 \, d\mathbf{x}}{\int |c(\mathbf{x})|^2 \, d\mathbf{x}} \right), \tag{2.3}$$

where the integrals are over the region of image space for which the quantities in the integrals are defined, unless otherwise stated. Note that this notation is often employed (in the interests of brevity) where the quantities are samples so that a summation is implied.

In a similar manner, the energy of the image $f(\mathbf{x})$ is defined as

$$Eng[f(\mathbf{x})] = \int |f(\mathbf{x})|^2 \, d\mathbf{x}. \tag{2.4}$$

2.2 COMPACT IMAGES

All real-world images are finite in size because the field of view of any real-world imaging instrument is finite. In addition, a real-world image $p(\mathbf{x})$, has finite amplitude, that is $|p(\mathbf{x})| < \infty$. An image having both finite size and finite amplitude is said to be *compact* (Bates and McDonnell, 1989).

The support of an image $p(\mathbf{x})$, denoted by $S_p(\mathbf{x})$, is the finite region of space it occupies; outside $S_p(\mathbf{x})$, the magnitude of $p(\mathbf{x})$ is small enough to be negligible, i.e. $|p(\mathbf{x})| \leq \epsilon$, where ϵ is a constant with small value. When $p(\mathbf{x})$ is a true image, $\epsilon = 0$; when $p(\mathbf{x})$ is recorded through an imaging instrument and thus contaminated, ϵ is a small positive constant related to the magnitude of the contamination.

A rectangular region with sides parallel to arbitrarily defined Cartesian coordinates and just enclosing the envelope of the support $S_p(\mathbf{x})$ is the *image box* of $p(\mathbf{x})$, denoted $B_p(\mathbf{x})$. The length of $B_p(\mathbf{x})$ in the k^{th} direction is defined as the extent of $p(\mathbf{x})$ in that direction, denoted by $L_p(x_k)$. Figure 2.1 shows the support, image box and extents of a two-dimensional image.

2.3 IMAGE FORM

The appearance or the *form* of an image is not fundamentally changed if the image is scaled, translated and even reflected (Bates and McDonnell, 1989, §20). The image form is a useful concept in blind deconvolution, because in this situation both $f(\mathbf{x})$ and $h(\mathbf{x})$ are unknown *a priori* and are estimated one from the other in conjunction with the given measurement $g(\mathbf{x})$ and any prior knowledge about the image. Clearly, for example, $f(\mathbf{x}) \odot h(\mathbf{x}) = f(-\mathbf{x}) \odot h(-\mathbf{x}) = af(\mathbf{x} - \mathbf{x}_0) \odot \frac{1}{a} h(\mathbf{x} + \mathbf{x}_0)$, where a is a nonzero constant

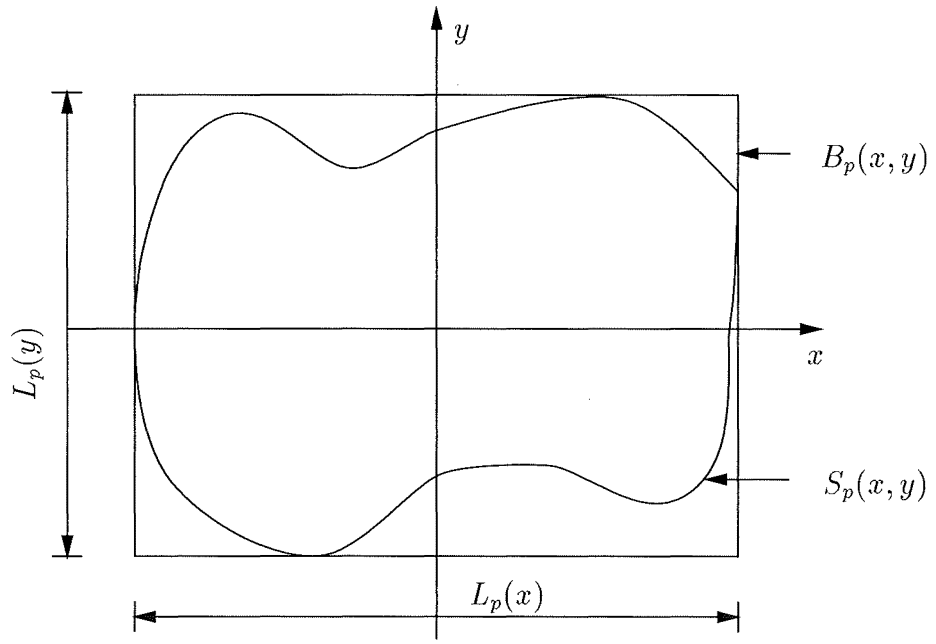


Figure 2.1 The support, image box and extent of a two-dimensional image

and $\mathbf{x}_0$ is an arbitrary location. Thus an estimate in blind deconvolution can only be of the form of $f(\mathbf{x})$.

2.4 FOURIER THEORY

This section introduces useful basic mathematical theory employed extensively in this thesis. The Fourier transform is described in §2.4.1. Some of its useful properties are described in §2.4.2. The theory of uniquely representing a continuous image by its discrete samples is discussed in §2.4.3. The discrete Fourier transform, necessary for performing an image restoration process on a digital computer, is introduced in §2.4.4.

2.4.1 The Fourier transform

The Fourier transform of an image $f(\mathbf{x})$ is defined (Bates and McDonnell 1989, §6) as

$$F(\mathbf{u}) = \int_{-\infty}^{\infty} f(\mathbf{x}) e^{i2\pi \mathbf{u}\cdot\mathbf{x}} d\mathbf{x}, \tag{2.5}$$

where $d\mathbf{x}$ is the K-dimensional volume element, $\mathbf{u}$ is the position vector representing *spatial frequency* and $\mathbf{u}\cdot\mathbf{x}$ is the scalar product of $\mathbf{u}$ and $\mathbf{x}$, that is

$$\mathbf{u}\cdot\mathbf{x} = \sum_{k=1}^{K} u_k x_k. \tag{2.6}$$

In the majority of cases considered in this thesis, $K = 2$. The inverse Fourier transform of $F(\mathbf{u})$ is defined by

$$f(\mathbf{x}) = \int_{-\infty}^{\infty} F(\mathbf{u}) e^{-i2\pi \mathbf{u}\cdot\mathbf{x}} d\mathbf{u}. \tag{2.7}$$

$F(\mathbf{u})$ is also known as the *spectrum* of $f(\mathbf{x})$.

The connection between the image space and the Fourier space is established by the expression

$$f(\mathbf{x}) \longleftrightarrow F(\mathbf{u}) \tag{2.8}$$

where $\longleftrightarrow$ identifies that $f(\mathbf{x})$ and $F(\mathbf{u})$ form a Fourier transform pair. Equation (2.8) emphasizes the invertible relationship existing between an image and its spectrum. The terminology of the Fourier transform and its inverse can be concisely expressed by the symbols $FT[\cdot]$ and $IFT[\cdot]$ respectively, thus (2.5) and (2.7) can be symbolically described as $F(\mathbf{u}) = FT[f(\mathbf{x})]$ and $f(\mathbf{x}) = IFT[F(\mathbf{u})]$.

The Fourier transform has its definite physical meaning in many practical situations. Typical examples are that a converging optical lens possesses the property of performing two-dimensional Fourier transformations (Goodman, 1968, ch. 5) and the light amplitude distribution at the front and back focal planes of the lens forms a Fourier transform pair.

2.4.2 Properties

2.4.2.1 Convolution

When a two-dimensional true image $f(\mathbf{x})$ is passed through an ideal linear system, the output image or the measurement is characterized by the convolution of $f(\mathbf{x})$ and the system response function $h(\mathbf{x})$. That is

$$\begin{aligned} b(\mathbf{x}) &= \int_{-\infty}^{\infty} f(\mathbf{x}')\, h(\mathbf{x} - \mathbf{x}')\, d\mathbf{x}' \\ &= f(\mathbf{x}) \odot h(\mathbf{x}). \end{aligned} \tag{2.9}$$

where the term $h(\mathbf{x} - \mathbf{x}')$ indicates that $h(\mathbf{x})$ is invariant for each point of $f(\mathbf{x})$ (i.e. isoplanatic) as discussed in §1.2. The functions $f(\mathbf{x})$ and $h(\mathbf{x})$ are said to be components of the convolution. If a function cannot be expressed as the convolution of two or more components, it is said to be *irreducible* (Lane and Bates, 1987b).

Equation (2.9) reveals that the extent of the convolution in each coordinate direction is greater than the extent of each of its components, but no greater than the summation of the extents of these components. That is

$$L_{f(\mathbf{x})\odot h(\mathbf{x})}(x_k) \leq L_f(x_k) + L_h(x_k), \tag{2.10}$$

where equality holds when the convolution components are positive functions. This is known as the extent of convolution theorem (Bates and M^{c}Donnell, 1989, §7).

A useful and important property of the convolution is the relationship between (2.9) and its Fourier transform, which is known as the *convolution theorem*, namely

$$f(\mathbf{x}) \odot h(\mathbf{x}) \longleftrightarrow F(\mathbf{u})H(\mathbf{u}). \tag{2.11}$$

The convolution theorem allows one to compute the result of convolution in image space by simple multiplication in Fourier space.

From (1.5) and (2.11), it is seen that the Fourier transform of a contaminated blurred image is described by

$$G(\mathbf{u}) = F(\mathbf{u})H(\mathbf{u}) + C(\mathbf{u}), \tag{2.12}$$

where $C(\mathbf{u})$ is the Fourier transform of $c(\mathbf{x})$.

2.4.2.2 Correlation

The correlation function is formed by integrating the lagged products of two images, $f(\mathbf{x})$ and $d(\mathbf{x})$, by

$$f(\mathbf{x}) * d(\mathbf{x}) = \int_{-\infty}^{\infty} f(\mathbf{x}')\, d(\mathbf{x}+\mathbf{x}')\, d\mathbf{x}', \tag{2.13}$$

where $*$ is the two-dimensional operator. As a special case when $f(\mathbf{x}) = d(\mathbf{x})$, their correlation is referred to as *autocorrelation* defined by

$$\begin{aligned} ff(\mathbf{x}) &= f^*(\mathbf{x}) * f(\mathbf{x}) \\ &= \int_{-\infty}^{\infty} f^*(\mathbf{x}')\, f(\mathbf{x}+\mathbf{x}')\, d\mathbf{x}', \end{aligned} \tag{2.14}$$

where $f^*(\mathbf{x})$ is the complex conjugate of $f(\mathbf{x})$. The Fourier transforms of Equations (2.13) and (2.14) give rise to the *correlation theorem* and *autocorrelation theorem* (Bates and McDonnell, 1989, §7), which are

$$f(\mathbf{x}) * d(\mathbf{x}) \longleftrightarrow F^*(\mathbf{u})\, D(\mathbf{u}) \tag{2.15}$$

and

$$f^*(\mathbf{x}) * f(\mathbf{x}) \longleftrightarrow |F(\mathbf{u})|^2. \tag{2.16}$$

2.4.2.3 Parseval's theorem

Parseval's theorem (Bracewell, 1978) or the energy conservation theorem (Bates and McDonnell 1989, §6) states that the energy of an image is equal to the energy of its spectrum. That is energy is preserved between the image space and the Fourier space and is expressed by

$$\int_{-\infty}^{\infty} |f(\mathbf{x})|^2 d\mathbf{x} = \int_{-\infty}^{\infty} |F(\mathbf{u})|^2 d\mathbf{u}. \tag{2.17}$$

2.4.3 Sampling

To process a continuous image in a digital computer, it is first necessary to convert the image into a discrete form. The continuous image is therefore sampled at a sequence of equispaced points, the value of the image at each sampled point taking the same as the value of the continuous image at that point. A two-dimensional sampled image, denoted $f_s(x, y)$, can be obtained by taking samples of a continuous image $f(x, y)$ at spatial intervals $\triangle_x$ and $\triangle_y$, that is

$$f_s(x, y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} f(n\triangle_x, m\triangle_y)\delta(x - n\triangle_x)\delta(y - m\triangle_y), \tag{2.18}$$

where $\triangle_x$ and $\triangle_y$ are called the *sampling intervals* in x and y directions of Cartesian coordinate. The reciprocal quantities $1/\triangle_x$ and $1/\triangle_y$ are the *sampling frequencies* in x and y directions respectively. When the sampling frequency in each direction is twice or greater than the maximum frequency of the image in the corresponding direction, the Nyquist sampling frequency, the continuous image can be uniquely determined by its sampled image (Bracewell, 1978). Since sampling an image causes its spectrum to be periodic, if the sampling frequency is less than Nyquist sampling frequency, each adjacent period of the periodic spectrum is overlapped, and information is irretrievably lost (Bracewell, 1978). This is known as *aliasing*. The smaller is the sampling frequency, the more is the overlapping and thus the more severe is the distortion.

2.4.4 Discrete Fourier transform

Many image processing algorithms described in this thesis require the computation of Fourier transform pairs. Such computation can be realised by utilizing the discrete Fourier transform (DFT). The DFT may be used to relate the samples of a compact image to the samples of its spectrum. For a compact image $f(x, y)$ with $L_f(x)$ and $L_f(y)$ as extent in x and y directions, the DFT of its sampled version can be obtained by substituting equation (2.18) into (2.5) with the upper and the lower bounds being replaced by finite numbers, thus

$$F(u_n, v_m) = \sum_{x_n=0}^{N-1} \sum_{y_m=0}^{M-1} f(x_n, y_m) \exp^{i2\pi(u_n x_n/N + v_m y_m/M)}, \tag{2.19}$$

and

$$f(x_n, y_m) = \frac{1}{NM} \sum_{u_n=0}^{N-1} \sum_{v_m=0}^{M-1} F(u_n, v_m) \exp^{-i2\pi(u_n x_n/N + v_m y_m/M)}, \tag{2.20}$$

where x_n, y_m and u_n, v_m are integers, e.g. x_n represents n^{th} image space sample in the x direction, N and M are the number of samples in x, u and y, v directions respectively of the two spaces with $N = L_f(x)/\triangle_x$ and $M = L_f(y)/\triangle_y$. Equations (2.19) and (2.20) reveal that the DFT is actually a trigonometric Fourier series representation (Bates and

McDonnell, 1989, §12). Since sampling in one domain corresponds to periodicity in the other, the DFT pair described by (2.19) and (2.20) are the expressions of one-period segments of periodic functions. The details of deriving the DFT from a continuous function can be found in Bates and McDonnell (1989) and Bracewell (1978).

The implementation of the DFT has been effectively sped up by invoking the *fast Fourier transform* (FFT) (Brigham, 1974). All computation of the Fourier transformation involved in this thesis is done by FFT. The properties of the continuous Fourier transform described in §2.4.2 are valid for both the DFT and the FFT (Oppenheim *et al.*, 1983, p336).

Throughout this thesis, a function $f(\mathbf{x})$ is used to represent both continuous and discrete quantities.

Chapter 3

REVIEW OF BLIND DECONVOLUTION METHODS

This chapter reviews various techniques available for solving blind deconvolution problems. These methods are classified as iterative methods, zero-based methods and optimization methods. Emphasis in the review is put on the iterative methods from which the new algorithms proposed in chapter 5 are developed. This review provides a survey of the existing methods and supplies groundwork for comparison of different methods presented in chapter 6. Two standard deconvolution methods are firstly introduced in §3.1, since they are the foundation of the iterative blind deconvolution methods addressed next. Iterative methods are described in §3.2. Zero-based blind deconvolution methods are discussed in §3.3 and optimization methods are described in §3.4.

3.1 STANDARD DECONVOLUTION METHODS

Enormous effort has been devoted to solving the standard deconvolution problems and numerous techniques have been developed (see for example Andrews and Hunt, 1977; Bates and McDonnell, 1989). However, only two of these techniques are discussed here. The principle and significance of these two methods on deconvolution are adopted and considered by the iterative blind deconvolution methods in §3.2 and the new algorithms presented in chapter 5. The two standard deconvolution methods are the inverse filter described in §3.1.1 and the Wiener filter discussed in §3.1.2.

3.1.1 Inverse filtering

The straightforward solution to the standard deconvolution problem is to invert the equation describing the spectrum of the blurred image in (2.11). The spectrum of the image $f(\mathbf{x})$ can be obtained by

$$F(\mathbf{u}) = B(\mathbf{u})I(\mathbf{u}), \tag{3.1}$$

where

$$I(\mathbf{u}) = \frac{1}{H(\mathbf{u})} \tag{3.2}$$

is known as the *inverse filter* or the *ideal multiplicative filter*. If $H(\mathbf{u})$ is known and the magnitude of $H(\mathbf{u})$ is not equal to zero for any frequency $\mathbf{u}$, $F(\mathbf{u})$ can be fully and precisely recovered. However, the performance of the inverse filter is limited by the following two severe problems:

Firstly, in the frequency region where the magnitude of $H(\mathbf{u})$, $|H(\mathbf{u})|$, is equal to or very close to zero, information about $|F(\mathbf{u})|$ at the related frequencies is missing from the convolution $B(\mathbf{u})$ and $I(\mathbf{u})$ is undefined. According to the convolution theorem described in (2.11), $|B(\mathbf{u})|$ is zero whenever $|F(\mathbf{u})|$ or $|H(\mathbf{u})|$ is zero. As it is most likely that $|H(\mathbf{u})|$ is zero at some frequencies, such as out-of-focus and linear motion types of psfs, there is no information about $F(\mathbf{u})$ being recorded in B$(\mathbf{u})$. On the other hand, when $|H(\mathbf{u})|$ is zero, the inverse filter $I(\mathbf{u})$ becomes infinitely large and undefined. $I(\mathbf{u})$ is, therefore, not able to be used to retrieve $F(\mathbf{u})$ even in the absence of convolution contamination.

Secondly, the restoration may be dominated by the inverse filtered contamination. Real-world images are inevitably contaminated. When the inverse filter is applied to the spectrum of a contaminated blurred image $g(\mathbf{x})$ (2.12), the spectrum of the restoration becomes

$$\hat{F}(\mathbf{u}) = G(\mathbf{u})I(\mathbf{u}) = F(\mathbf{u}) + \frac{C(\mathbf{u})}{H(\mathbf{u})}. \tag{3.3}$$

If $|H(\mathbf{u})|$ is very small at some frequencies, the contamination related term in (3.3) can be significantly greater than the spectrum of the true image, that is

$$|C(\mathbf{u})|/|H(\mathbf{u})| \gg |F(\mathbf{u})|. \tag{3.4}$$

Contamination $C(\mathbf{u})$ is, thus, overexaggerated by the inverse filter and dominates in the restored spectrum $\hat{F}(\mathbf{u})$ (Bates and McDonnell, 1989, §16). Such circumstance usually occurs in high frequency region. This is because the magnitude of the spectrum of a blurred image is typically highest at low frequencies and decreases very rapidly with increasing frequency, whilst the magnitude of the spectrum of a contamination function tends to have even variation for all frequencies. The outcome of $\hat{F}(\mathbf{u})$ being dominated by contamination at high frequencies is that in the image space the restored image $f(\mathbf{x})$ is severely contaminated and supplies worthless information (Biemond *et al.*, 1990). Even if the contamination of the blurred image is small (for example SNR = 40dB), the resulting inverse filter solution is still significantly dominated by the amplified contamination. The high sensitivity of the restoration to the measurement error using the inverse filter is an example that deconvolution problem is an ill-conditioned problem (Tikhonov and Arsenin, 1977; Nashed, 1981).

Consider on the other hand, when (3.4) is satisfied,

$$|F(\mathbf{u})H(\mathbf{u})| \ll |C(\mathbf{u})|. \tag{3.5}$$

The spectrum of the contaminated blurred image, $G(\mathbf{u})$, is dominated by the contamination $C(\mathbf{u})$. Information about the spectrum of ideal convolution $F(\mathbf{u})$ $H(\mathbf{u})$ in $G(\mathbf{u})$ is

swamped by the contamination $C(\mathbf{u})$. It is therefore generally impossible to retrieve $F(\mathbf{u})$ from $G(\mathbf{u})$ at the frequencies where (3.5) is met.

The above two limitations on the usefulness of the inverse filter are to some extent shared by all methods of deconvolution. Other filters, such as the Wiener filter which is discussed next, substantially reduce the effect of the limitations.

3.1.2 Wiener filtering

The effect of the limitations of the inverse filter can be reduced to a much less noticeable degree by its modification, the *Wiener filter* (the *least squares filter*), defined by

$$W(\mathbf{u}) = \frac{H^*(\mathbf{u})}{|H(\mathbf{u})|^2 + \alpha(\mathbf{u})}, \tag{3.6}$$

where $\alpha(\mathbf{u})$ is a measure of the noise-to-signal ratio, defined by

$$\alpha(\mathbf{u}) = \frac{|C(\mathbf{u})|^2}{|F(\mathbf{u})|^2}. \tag{3.7}$$

The spectrum of the restoration can be obtained by

$$\hat{F}(\mathbf{u}) = W(\mathbf{u})G(\mathbf{u}). \tag{3.8}$$

It is shown that the Wiener filter (3.6) is formed in such a manner that the least squared error between the true image $f(\mathbf{x})$ and its restoration $\hat{f}(\mathbf{x})$ is minimized, provided $c(\mathbf{x})$ is uncorrelated with $f(\mathbf{x})$ and either of them has zero mean (Bates and M^{c}Donnell, 1989, §16, Rosenfeld and Kak, 1982).

The Wiener filter performs as the inverse filter (3.2) at frequencies where $|F(\mathbf{u})H(\mathbf{u})|$ is substantially greater than $|C(\mathbf{u})|$, or equivalently where $|H(\mathbf{u})|^2 \gg \alpha(\mathbf{u})$. The Wiener filter also compensates for the problems suffered by the inverse filter in an appropriate and simple manner. For frequencies at which $|H(\mathbf{u})| = 0$, $|W(\mathbf{u})| = 0$. While this does not allow the restoration of $F(\mathbf{u})$ at these frequencies to be recovered, at least $\hat{F}(\mathbf{u})$ is well behaved. At frequencies where $G(\mathbf{u})$ is dominated by $C(\mathbf{u})$, that is (3.4) is satisfied, or equivalently $|H(\mathbf{u})|^2 \ll \alpha(\mathbf{u})$, the magnitude of the Wiener filter is suppressed to a small value of $|W(\mathbf{u})| \approx |H(\mathbf{u})|/\alpha(\mathbf{u})$, while the phase of $W(\mathbf{u})$ is maintained the same as that of the inverse filter (3.2). The influence of the dominated convolution contamination on the restoration of $F(\mathbf{u})$ is reduced. The contamination amplification suffered by the inverse filter is therefore prevented.

In practice, however, an accurate estimate of the noise-to-signal ratio needed for $\alpha(\mathbf{u})$ is rarely available. Various attempts have been made to estimate this ratio (Hall, 1979). One of the common approaches is to approximate $\alpha(\mathbf{u})$ to a constant α, which is known as the Wiener filter constant (Rosenfeld and Kak, 1982; Bates and McDonnell, 1989). As a consequence, equation (3.6) becomes

$$W(\mathbf{u}) = \frac{H^*(\mathbf{u})}{|H(\mathbf{u})|^2 + \alpha}. \tag{3.9}$$

From the above analysis, it is obvious that a suitable choice of α is around the mean value of $\alpha(\mathbf{u}) = |C(\mathbf{u})|^2/|F(\mathbf{u})|^2$. In practice, however, the choice of α is usually determined experimentally. A large range of values of α are tested, the one resulting the "best" restoration is considered to be suitable.

A limitation of the Wiener filter is that it attenuates the effect of contamination at the expense of its ability of deblurring (Biemond *et al.*, 1990). The involvement of its filter constant in the filter introduces bias to the ideal deconvolution operator. Useful high frequency information of the image is removed or weakened due to the presence of the filter constant. This limitation is further addressed in §3.2.3 and §5.1 in terms of the effect of the modified Wiener filter in blind deconvolution.

It is noticed that in many applications the conditions for which the Wiener filter is optimal in a least squares sense are not always satisfied; also $\alpha(\mathbf{u})$ is generally not known. Despite this, faithful restorations can still be obtained by employing the Wiener filter. It is therefore suggested by Bates *et al.* (1984) to recognize the Wiener filter merely as a modified inverse filter, which ensures the numerical stability of deconvolution. This point is further discussed in §5.3.4.

3.2 ITERATIVE TECHNIQUES

Iterative blind deconvolution algorithms are typically based on an iterative loop which alternates between image and Fourier spaces. Various constraints and prior knowledge regarding the image and the psf are allowed to be incorporated into the loop, in an attempt to urge the algorithm to converge towards its solution.

This section describes three iterative blind deconvolution algorithms. In §3.2.1 an iterative technique for positive images is introduced. This method consists of two separate steps of retrieving the phases and magnitudes of $F(\mathbf{u})$ and $H(\mathbf{u})$. Sections 3.2.2 and 3.2.3 describe two iterative algorithms proposed by Ayers and Dainty (1988) and Davey *et al.* (1989) respectively. The basic loop of these two algorithms are extended from that of the iterative phase retrieval algorithms (Gerchberg and Saxton, 1972; Fienup, 1982).

3.2.1 Phase based method

Lane and Bates (1987b) (*cf.* Bates and Lane, 1987; Lane, 1988) propose an iterative algorithm which firstly employs the spectral magnitude of a given convolution $g(\mathbf{x})$ to retrieve Fourier phases of the image $f(\mathbf{x})$ and the psf $h(\mathbf{x})$, and then uses the Fourier phases of $f(\mathbf{x})$ and $h(\mathbf{x})$ to restore their Fourier magnitudes. The restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ are generated by inverse Fourier transforming their spectral estimates, respectively.

Consider that when $g(\mathbf{x})$ is uncontaminated, $g(\mathbf{x})$ shares the same spectral magnitude $|G(\mathbf{u})|$ with the following three quantities,

$$G_2(\mathbf{u}) = F^*(\mathbf{u})H^*(\mathbf{u}), \tag{3.10a}$$

$$G_3(\mathbf{u}) = F^*(\mathbf{u})H(\mathbf{u}), \tag{3.10b}$$

$$G_4(\mathbf{u}) = F(\mathbf{u})H^*(\mathbf{u}). \tag{3.10c}$$

It is obvious that the phases of $G_3(\mathbf{u})$ and $G_4(\mathbf{u})$ are significantly different from the phase of $G(\mathbf{u})$.

The first step of the phase-based algorithm is to compute the spectral magnitude of the given convolution, $|G(\mathbf{u})|$. A phase retrieval algorithm (e.g. that of Fienup, 1982) is then applied to $|G(\mathbf{u})|$ to restore an estimate $\hat{G}(\mathbf{u})$ which corresponds to either $G_3(\mathbf{u})$ or $G_4(\mathbf{u})$. This can be done by repeating the phase retrieval with different starting images until the resulting phase of $\hat{G}(\mathbf{u})$ is significantly different from the phase of $G(\mathbf{u})$.

The next step is to compute the spectral phases of $f(\mathbf{x})$ and $h(\mathbf{x})$. Adding and subtracting the phases of $G(\mathbf{u})$ and $\hat{G}(\mathbf{u})$ produce twice the phases of $F(\mathbf{u})$ and $H(\mathbf{u})$. Thus, assuming $\hat{G}(\mathbf{u}) = G_3(\mathbf{u})$,

$$\begin{aligned} \mathcal{P}[G(\mathbf{u})] + \mathcal{P}[\hat{G}(\mathbf{u})] &= 2\mathcal{P}[H(\mathbf{u})] \text{ modulo } 2\pi \\ &= 2\mathcal{P}[H(\mathbf{u}) \pm \pi] \text{ modulo } 2\pi, \end{aligned} \tag{3.11}$$

$$\begin{aligned} \mathcal{P}[G(\mathbf{u})] - \mathcal{P}[\hat{G}(\mathbf{u})] &= 2\mathcal{P}[F(\mathbf{u})] \text{ modulo } 2\pi \\ &= 2\mathcal{P}[F(\mathbf{u}) \pm \pi] \text{ modulo } 2\pi. \end{aligned} \tag{3.12}$$

In practice, it is impossible to know which phase is for $F(\mathbf{u})$ and which is for $H(\mathbf{u})$ since there is no way to identify whether $\hat{G}(\mathbf{u})$ is equal to $G_3(\mathbf{u})$ or $G_4(\mathbf{u})$. Furthermore, equations (3.11) and (3.12) only give phases of $F(\mathbf{u})$ and $H(\mathbf{u})$ modulo π rather than the true phases of $F(\mathbf{u})$ and $H(\mathbf{u})$.

The final step of the phase based blind deconvolution algorithm is to retrieve the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ independently from $\mathcal{P}[F(\mathbf{u})]$ modulo π and $\mathcal{P}[H(\mathbf{u})]$ modulo π. Correctness of the support of $f(\mathbf{x})$ and $h(\mathbf{x})$ is crucial in this final process since the phases of $F(\mathbf{u})$ and $H(\mathbf{u})$ contain information of the location of $f(\mathbf{x})$ and $h(\mathbf{x})$. This process is typically repeatedly applied with different support estimates of $f(\mathbf{x})$ or $h(\mathbf{x})$ until the correct support estimate is found and the algorithm has converged.

Lane and Bates (1987b) and Lane (1988, §6.3) have demonstrated examples of applying their algorithm on contaminated blurred images.

3.2.2 Ayers and Dainty algorithm

Ayers and Dainty (1988) propose a blind deconvolution algorithm for iteratively retrieving both estimates of the image $f(\mathbf{x})$ and psf $h(\mathbf{x})$ simultaneously. The main feature of

their algorithm is to generate a new estimate using deconvolution principle in the Fourier space and force its inverse Fourier transform to comply with known constraints in the image space. Both estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ are alternately produced in this manner at each iteration and the algorithm is expected to converge to its solution thereby.

The basic structure of the iterative processing loop is described in Figure 3.1. For a given contaminated blurred image $g(\mathbf{x})$, the operation of the algorithm starts with an estimate of $f(\mathbf{x})$, denoted by $\hat{f}_i(\mathbf{x})$, where the subscript i stands for the loop counter and the hat is employed to indicate that the estimate complies with the image space constraints. The Fourier transform of $\hat{f}_i(\mathbf{x})$ is combined with $G(\mathbf{u})$ to generate $H_i(\mathbf{u})$, which is the new spectral estimate of $h(\mathbf{x})$ at i^{th} iteration. The inverse Fourier transform of $H_i(\mathbf{u})$ is then forced to comply with the image space constraints, which produces an improved estimate of $h(\mathbf{x})$ at the iteration, denoted by $\hat{h}_i(\mathbf{x})$. $\hat{h}_i(\mathbf{x})$ and $g(\mathbf{x})$ are used in the second half of the loop to produce $f_i(\mathbf{x})$ in a similar manner as $\hat{f}_i(\mathbf{x})$ and $g(\mathbf{x})$ are employed to yield $h_i(\mathbf{x})$. The image space constraints are enforced on $f_i(\mathbf{x})$ to yield $\hat{f}_{i+1}(\mathbf{x})$ which is an improved estimate of $f(\mathbf{x})$ and used as an input estimate for the next iteration. The initial estimate of $f(\mathbf{x})$, $\hat{f}_0(\mathbf{x})$, is usually chosen to have random values for all pixels within the image support.

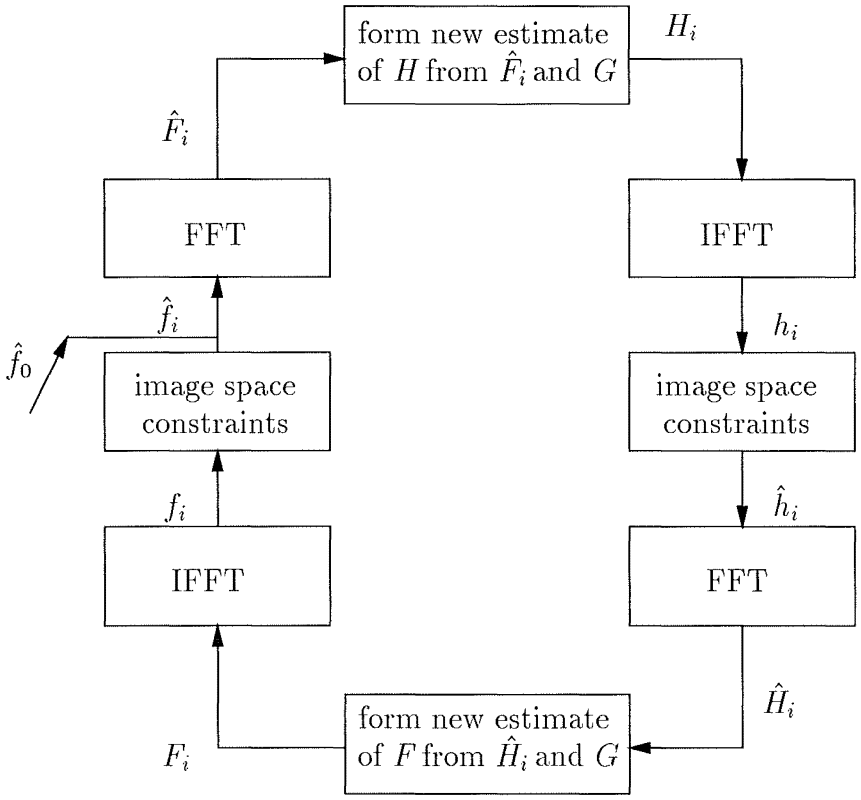


Figure 3.1 The basic loop of the iterative blind deconvolution algorithm proposed by Ayers and Dainty.

The basic steps of the algorithm can be mathematically described as

$$\hat{F}_i(\mathbf{u}) = FFT[\hat{f}_i(\mathbf{x})], \tag{3.13a}$$

$$H_i(\mathbf{u}) = Fourier\_space\_constraint[G(\mathbf{u}), \hat{F}_i(\mathbf{u})], \tag{3.13b}$$

$$h_i(\mathbf{x}) = IFFT[H_i(\mathbf{u})], \tag{3.13c}$$

$$\hat{h}_i(\mathbf{x}) = \begin{cases} h_i(\mathbf{x}) + \chi_i^h & \text{for } \mathbf{x} \notin \Omega_i^h, \\ \chi_i^h & \text{for } \mathbf{x} \in \Omega_i^h, \end{cases} \tag{3.13d}$$

$$\hat{H}_i(\mathbf{u}) = FFT[\hat{h}_i(\mathbf{x})], \tag{3.13e}$$

$$F_i(\mathbf{u}) = Fourier\_space\_constraint[G(\mathbf{u}), \hat{H}_i(\mathbf{u})], \tag{3.13f}$$

$$f_i(\mathbf{x}) = IFFT[F_i(\mathbf{u})], \tag{3.13g}$$

$$\hat{f}_{i+1}(\mathbf{x}) = \begin{cases} f_i(\mathbf{x}) + \chi_i^f & \text{for } \mathbf{x} \notin \Omega_i^f, \\ \chi_i^f & \text{for } \mathbf{x} \in \Omega_i^f, \end{cases} \tag{3.13h}$$

where Ω_i^h and Ω_i^f are the sets of pixels at which $h_i(\mathbf{x})$ and $f_i(\mathbf{x})$ violate the image space constraints for $h(\mathbf{x})$ or $f(\mathbf{x})$ respectively, and where χ_i^h and χ_i^f are real constants related to energy preservation of the two estimates separately, more detail about the energy preservation is addressed in the following.

The image space constraints employed by Ayers and Dainty are positivity and energy preservation. A negative value of the estimate $h_i(\mathbf{x})$ of (3.13c) or $f_i(\mathbf{x})$ of (3.13g) at any pixel is considered to violate the positivity constraint and is replaced by zero. Summation of the negative values of the estimate, $\chi_i^h N$ or $\chi_i^f N$ (N is the total number of pixels of $h_i(\mathbf{x})$ or $f_i(\mathbf{x})$), is evenly redistributed over the estimate so as to maintain the summation of the pixel's values of the estimate before and after applying the positivity constraint. The process is repeated until no pixel of the constrained estimate contains negative value.

The Fourier space constraint is realised by forcing the product of the two estimates, $\hat{F}_i(\mathbf{u})$ and $H_i(\mathbf{u})$, or $F_i(\mathbf{u})$ and $\hat{H}_i(\mathbf{u})$, to be equal to $G(\mathbf{u})$. Applying this constraint produces the new spectral estimate $H_i(\mathbf{u})$ (3.13b) or $F_i(\mathbf{u})$ (3.13f) at each iteration, which corresponds to the deconvolution process in the image space. The simplest way to apply the Fourier space constraint and therefore to generate the new estimate is by invoking an inverse filter (§3.1.1). Taking account the two practical difficulties when performing an inverse filter: one is its error sensitive nature encountered when the magnitude of the denominator of the filter is very low; the other is the lack of information about the ideal convolution spectrum $|F(\mathbf{u})|\,|H(\mathbf{u})|$ contained in $G(\mathbf{u})$ when either $|F(\mathbf{u})|$ or $|H(\mathbf{u})|$ is equal to zero (*cf.* §3.1.1), Ayers and Dainty (1988) present a composite form of the Fourier space constraint to deal with the limitations of the inverse filter, which

is described in the following when $F_i(\mathbf{u})$ is to be generated,

$$F_i(\mathbf{u}) = \begin{cases} \hat{F}_i(\mathbf{u}) & \text{if } |G(\mathbf{u})| \leq cl, \\ (1-\eta)\hat{F}_i(\mathbf{u}) + \eta\dfrac{G(\mathbf{u})}{\hat{H}_i(\mathbf{u})} & \text{if } |G(\mathbf{u})| > cl \quad \text{and} \quad |\hat{H}_i(\mathbf{u})| \geq |G(\mathbf{u})|, \\ \dfrac{1}{\dfrac{1-\eta}{\hat{F}_i(\mathbf{u})} + \dfrac{\eta\hat{H}_i(\mathbf{u})}{|G(\mathbf{u})|}} & \text{if } |G(\mathbf{u})| > cl \quad \text{and} \quad |\hat{H}_i(\mathbf{u})| < |G(\mathbf{u})|, \end{cases} \tag{3.14}$$

where η is a constant with $0 < \eta \leq 1$, and cl is an estimate of the contamination level in $G(\mathbf{u})$. Equation (3.14) shows when $|G(\mathbf{u})| \leq cl$, $G(\mathbf{u})$ supplies no useful information of $F(\mathbf{u})$ and $H(\mathbf{u})$, the new estimate $F_i(\mathbf{u})$ retains its previous estimate $\hat{F}_i(\mathbf{u})$ whose inverse Fourier transform satisfies the image space constraints; when $|G(\mathbf{u})| > cl$, $F_i(\mathbf{u})$ is formed by averaging both $\hat{F}_i(\mathbf{u})$ and $G(\mathbf{u})/\hat{H}_i(\mathbf{u})$ in some manner that the effect of the small values of $|\hat{H}_i(\mathbf{u})|$ is considered.

Ayers and Dainty (1988) also introduce an apodization function, denoted by $A(\mathbf{u})$, to cope with the problem of extended regions of low or zero value in the convolution spectrum at high frequencies. Due to the nature of a typical convolution spectrum $G(\mathbf{u})$, often there is a falling off of $|G(\mathbf{u})|$ in high frequency region, or $G(\mathbf{u})$ is band limited when it is formed by a band-limited imaging system. $A(\mathbf{u})$ is therefore designed to be greater than zero for all frequencies up to some band limit and zero beyond. Its inverse Fourier transform satisfies the positivity constraint in the image space. $A(\mathbf{u})$ is implemented in the following manner: to band limit the newly generated spectral estimate $H_i(\mathbf{u})$ (3.13b) or $F_i(\mathbf{u})$ (3.13f), the estimate is multiplied by $A(\mathbf{u})$. To compensate the involvement of $A(\mathbf{u})$, the spectral estimate $\hat{F}_i(\mathbf{u})$ (3.13a) or $\hat{H}_i(\mathbf{u})$ (3.13e), whose inverse Fourier transform is just enforced by the image space constraint, is divided by $A(\mathbf{u})$.

The iterative blind deconvolution algorithm proposed by Ayers and Dainty (1988) uses more information of the spectrum of the given measurement $g(\mathbf{x})$ compared with the phase-based method in §3.2.1. Both magnitude and phase of $G(\mathbf{u})$ are used by the former while only the magnitude of $G(\mathbf{u})$ is employed in the latter. Ayers and Dainty demonstrated their algorithm to blindly deconvolve two uncontaminated positive images.

3.2.3 Davey algorithm

Davey *et al* (1989) extend the Ayers and Dainty algorithm (1988) in the manner which the image and Fourier space constraints are applied. In the image space, Davey *et al* (1989) not only use the positivity constraint (if it is applicable) as Ayers and Dainty (1988) do, they also employ a support constraint. In the Fourier space, they apply a Wiener-like filter as main part of its constraint to generate each new spectral estimate $H_i(\mathbf{u})$ or $F_i(\mathbf{u})$. Otherwise the basic structure of their algorithm remains the same as Ayers and Dainty's in Figure 3.1. This algorithm is called the *Davey Algorithm*

henceforth.

Since energy preservation applied by Ayers and Dainty (1988) is not used by Davey *et al* (1989), (3.13d) and (3.13h) are replaced by

$$\hat{h}_i(\mathbf{x}) = \begin{cases} h_i(\mathbf{x}) & \text{for } \mathbf{x} \notin \Omega_i^h, \\ 0 & \text{for } \mathbf{x} \in \Omega_i^h, \end{cases} \tag{3.15}$$

and

$$\hat{f}_{i+1}(\mathbf{x}) = \begin{cases} f_i(\mathbf{x}) & \text{for } \mathbf{x} \notin \Omega_i^f, \\ 0 & \text{for } \mathbf{x} \in \Omega_i^f, \end{cases} \tag{3.16}$$

respectively in the Davey Algorithm.

As an additional image space constraint, the support constraint supplies more information about $f(\mathbf{x})$ and $h(\mathbf{x})$ to their estimates and forces the estimates more vigorously towards their solutions than the positivity constraint alone does.

The Wiener-like filter, used in the Fourier space constraint of (3.13b), is defined by

$$W_{\hat{F}_i}(\mathbf{u}) = \frac{\hat{F}_i^*(\mathbf{u})}{|\hat{F}_i(\mathbf{u})|^2 + \beta / |\hat{F}_i(\mathbf{u})|^n}, \tag{3.17}$$

where β is a real positive constant and n is the exponent which is chosen as $n = 2$ by Davey *et al* (1989). The new spectral estimate of $h(\mathbf{x})$ at i^{th} iteration is generated by the Fourier space constraint of

$$H_i(\mathbf{u}) = W_{\hat{F}_i}(\mathbf{u}) G(\mathbf{u}). \tag{3.18}$$

Exchanging the role of $\hat{F}_i(\mathbf{u})$ in (3.17) for $\hat{H}_i(\mathbf{u})$, the Wiener-like filter associated with (3.13f) is given by

$$W_{\hat{H}_i}(\mathbf{u}) = \frac{\hat{H}_i^*(\mathbf{u})}{|\hat{H}_i(\mathbf{u})|^2 + \beta / |\hat{H}_i(\mathbf{u})|^n}, \tag{3.19}$$

and the new spectral estimate of $f(\mathbf{x})$ in (3.13f) is obtained by

$$F_i(\mathbf{u}) = W_{\hat{H}_i}(\mathbf{u}) G(\mathbf{u}). \tag{3.20}$$

The form of the Wiener-like filter in (3.19) is similar to that of the conventional Wiener filter (3.9). Davey (1989, §7.4) points out that the modification of scaling the filter constant β by $|\hat{H}_i(\mathbf{u})|^n$ in (3.19) is motivated by the desire to reduce the bias existing in the conventional Wiener filter (3.6). When $|\hat{H}_i(\mathbf{u})|$ is small the filter has more effect by such scaling; when $|\hat{H}_i(\mathbf{u})|$ is large, $\beta/|\hat{H}_i(\mathbf{u})|$ is small, so the bias introduced by the Wiener filter is reduced. The role of the Wiener-like filter (3.19) as the key part of the Fourier space constraint is further discussed in §4.2.

To evaluate the performance of the algorithm, it is ideal to measure the difference between the estimate and its true version during the iterative process, which is named

the *true error*. A normalized true error of an image estimate at i^{th} iteration is defined by

$$E_T[f_i] = \frac{\int_{\mathbf{x}} |f(\mathbf{x}) - f_i(\mathbf{x})|^2 \, d\mathbf{x}}{\int_{\mathbf{x}} |f(\mathbf{x})|^2 \, d\mathbf{x}}. \tag{3.21}$$

For brivity, this will be referred to as the true error henceforth. In practice, to evaluate $E_T[f_i]$ a summation is performed rather than integration implied by (3.21). When the algorithm is applied to deconvolve real world images, the true images are not available, the performance of the algorithm is often assessed by some form of objective error measures. One of them measures the amount that the estimate violates the image space constraints, which is called the *image space error*. A normalized image space error of $f_i(\mathbf{x})$ is described by

$$E_I[f_i] = \frac{\int_{\mathbf{x} \in \Omega_i^f} |f_i(\mathbf{x})|^2 \, d\mathbf{x}}{\int_{\mathbf{x}} |f_i(\mathbf{x})|^2 \, d\mathbf{x}}. \tag{3.22}$$

For simplicity, equation (3.22) will be referred to as the image space error thereafter.

Davey *et al* (1989) present examples of blindly deconvolving both contaminated positive and complex images and have demonstrated the superiority of their algorithm over the Ayers and Dainty algorithm (1988).

An obvious advantage of Davey *et al* algorithm is the imposition of the support constraint which gives additional information of the image form to its estimate and therefore forces the convergence of the algorithm more firmly. However, it should be noted that for the Ayers and Dainty algorithm there is no bias involved in its Fourier space constraint in (3.14). Further investigation regarding the factors which influence the performance of the Davey Algorithm, the role of the Wiener-like filter and the objective error measures to monitor the process are presented in chapter 4.

3.3 ZERO-BASED METHODS

When one member of a Fourier transform pair is compact, the other member may be represented by an entire function of exponential type (Requicha, 1980; Paley and Wiener, 1934). A property of entire functions of exponential type is that they are completely characterized, apart from a complex scaling constant, by their zeros (Levin, 1964). Thus a compact image is completely specified by the positions at which its spectrum is zero, which is the concept that the zero-based deconvolution methods are grounded upon. According to the convolution theorem described in (2.11), the spectrum $G(\mathbf{u})$ equals zero whenever the spectrum $F(\mathbf{u})$ or $H(\mathbf{u})$ is zero. Therefore, the zeros of $G(\mathbf{u})$ comprise the union of zeros of $F(\mathbf{u})$ and $H(\mathbf{u})$, provided that $g(\mathbf{x})$ is compact and uncontaminated. When $g(\mathbf{x})$ is a two (or more) dimensional compact image and the components of $\mathbf{u}$ are allowed to take complex values, all zeros of $G(\mathbf{u})$ form a

continuous and analytic surface, named the *zero-sheet* and defined $\mathcal{Z}[G(\mathbf{u})]$ (Lane and Bates, 1987a). If $f(\mathbf{x})$ and $h(\mathbf{x})$ are irreducible (*cf.* §2.4.2.1), zero-sheets of $F(\mathbf{u})$ and $H(\mathbf{u})$ contained in $\mathcal{Z}[G(\mathbf{u})]$ are continuous, distinct and separable. Restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ can then be retrieved using $\mathcal{Z}[F(\mathbf{u})]$ and $\mathcal{Z}[H(\mathbf{u})]$ respectively. Blind deconvolution is therefore feasible in principle for two or higher dimensional images. Blind deconvolution involving partitioning $\mathcal{Z}[F(\mathbf{u})]$ and $\mathcal{Z}[H(\mathbf{u})]$ have been realized by Lane and Bates (1987a) and Ghiglia *et al.* (1993). Both methods employ the analytic nature of zero-sheets $\mathcal{Z}[F(\mathbf{u})]$ and $\mathcal{Z}[H(\mathbf{u})]$. When the convolution $g(\mathbf{x})$ is contaminated, however, $g(\mathbf{x})$ is essentially always irreducible (Sanz and Huang, 1985). The addition of contamination on the zero-sheet of $G(\mathbf{u})$ links two zero-sheets of $F(\mathbf{u})$ and $H(\mathbf{u})$ at some discrete points. The zeros of $G(\mathbf{u})$ form a single and continuous sheet which no longer comprises two distinct smooth surfaces of $\mathcal{Z}[F(\mathbf{u})]$ and $\mathcal{Z}[H(\mathbf{u})]$. Techniques which Lane and Bates (1987a) and Ghiglia *et al.* (1993) used for separating zero-sheets of $F(\mathbf{u})$ and $H(\mathbf{u})$ when $g(\mathbf{x})$ is ideally blurred are no longer effective. Bates *et al.* (1990b) suggested that the addition of contamination to images may not prevent zero-based deconvolution methods from achieving useful results. This prediction has been realised in solving ensemble blind deconvolution problem and conventional deconvolution problem using contaminated measurements (Bones *et al.*, 1995). Satherley and Bones (1994) (*cf.* Satherley, 1994) propose a method for solving two-dimensional ensemble blind deconvolution problems. Their method considers *zero-tracks* (i.e. two-dimensional projections of zero-sheets) representing an ensemble of two-dimensional images. From the ensemble of zero-tracks calculated from the given contaminated ensemble blurred images, the algorithm extracts and processes the common tracks corresponding to the true image $f(\mathbf{x})$ and then reconstructs the estimate of the image $f(\mathbf{x})$. In a conventional deconvolution problem, an estimate of $h(\mathbf{x})$, $\hat{h}(\mathbf{x})$, is known. Watson *et al.* (1992) and Parker and Bones (1992a), respectively, utilize the knowledge of zero-sheet of $\hat{H}(\mathbf{u})$ and employ different techniques to separate the zero-sheet of $H(\mathbf{u})$ from zero-sheet of $G(\mathbf{u})$. The remaining parts of the zero-sheet of $G(\mathbf{u})$ is then assumed to belong to the zero-sheet of $F(\mathbf{u})$ which is then used to reconstruct the estimate of the image $f(\mathbf{x})$.

Recovering an image from its zero-sheet is an important task in zero based deconvolution. In one dimension, this image recovery is straightforward since the zeros of $F(\mathbf{u})$ are isolated points and finite in number (Lane and Bates, 1987a). However, in two dimensions, there are an infinite number of points involved in a zero-sheet and used to describe the image. This complicates the process of image recovery since only a finite number of these points is required to form the image. There are two major approaches to recover an image from its zero-sheet. One of the techniques uses all the zero points required for an inverse discrete Fourier transform to recover the image (Lane and Bates, 1987a; Lane *et al.*, 1987; Satherley and Bones, 1994). This method is simple but very sensitive to error in the locations of the selected zeros. Satherley and Parker (1993) and Bones *et al.* (1995) extend the method to allow a wide choice of zero-sheet sam-

ples to be used in the restoration of the image. The second method employs a linear equations approach. This approach uses a set of zero points to formulate a set of linear equations with the image pixels as the unknowns (Curtis *et al.*, 1985; Izraelevitz and Lim, 1987; Watson *et al.*, 1992).

One of the major advantages of zero-sheet based deconvolution is that the methods do not restrict the image to be positive or real. However the methods are highly computationally expensive and sensitive to contamination.

3.4 OPTIMIZATION METHODS

Posing a blind deconvolution problem as a minimization problem, the blind deconvolution problem can be approached using various well established optimization methods. The blind deconvolution problem can be presented as a minimization problem by:

> "Finding estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$, denoted $\tilde{f}(\mathbf{x})$ and $\tilde{h}(\mathbf{x})$, to minimize a selected error metric E."
>
> E is an objective quantity which incorporates the given measurement $g(\mathbf{x})$ and the estimates $\tilde{f}(\mathbf{x})$ and $\tilde{h}(\mathbf{x})$ in a manner to give a measure of how well $\tilde{f}(\mathbf{x})$ and $\tilde{h}(\mathbf{x})$ are recovered.

If the minimization problem is subjected to any constraint upon the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$, it is known as a constrained minimization problem. Otherwise it is called the unconstrained minimization problem.

This section introduces diverse blind deconvolution techniques using optimization methods. Blind deconvolution using simulated annealing is described in §3.4.1. The application of the conjugate gradient method to blind deconvolution is reviewed in §3.4.2. The maximum likelihood methods are described in §3.4.3 and the maximum entropy methods are mentioned in §3.4.4.

3.4.1 Simulated annealing method

The simulated annealing method is based on the Metropolis Algorithm (Metropolis *et al.*, 1953) devised for calculating averages of state functions in statistical mechanics. Pincus (1970) extend the Metropolis Algorithm to a global optimization method in the manner that the Metropolis Algorithm is operated repeatedly with a succession of decreasing values of the parameter T. Kirkpatrick *et al* (1983) popularize the method and term it as *simulated annealing*. Following Nieto Vesperinas and Bendez (1986), Nieto Vesperinas *et al.* (1988) and Navarro *et al.* (1989), who apply the simulated annealing technique to the phase retrieval problem, McCallum (1990b) (*cf.* McCallum, 1990a) extends this minimization technique to solve the blind deconvolution problem.

McCallum selected the error metric as

$$E[g] = \frac{\int_{\mathbf{x}} |g(\mathbf{x}) - \tilde{f}(\mathbf{x}) \odot \tilde{h}(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x}}{\int_{\mathbf{x}} |g(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x}}. \tag{3.23}$$

When applying the simulated annealing technique to the blind deconvolution problem, a pixel $\mathbf{x}'$ of the estimate $\tilde{f}(\mathbf{x})$ or $\tilde{h}(\mathbf{x})$ is displaced or perturbed by a quantity v, which is pseudo-randomly distributed in the range $[-\kappa, \kappa]$. κ is termed as the scale of perturbation. The change in the error $E[g]$ resulted from this perturbation or displacement of the pixel's value at $\mathbf{x}'$ is then calculated as δE. The perturbation is accepted unconditionally (i.e. the new value v at $\mathbf{x}'$ replaces the old one) if $\delta E \leq 0$; if $\delta E > 0$, the perturbation is accepted with probability $e^{-\delta E/T}$ where T is a control parameter known as temperature; if the perturbation is not accepted, the previous value at $\mathbf{x}'$ is retained. One image scan constitutes sequentially perturbing each pixel of the estimate of $f(\mathbf{x})$ or $h(\mathbf{x})$ in this manner. Image scans are repeated until equilibrium is reached. That occurs when the average number of perturbations resulting in $\delta E > 0$ is sufficiently close to the average number of perturbations resulting in $\delta E \leq 0$. Once equilibrium is reached, the values of both T and κ are decreased according to various prescriptions. The above procedure, which constitutes one cycle, is then repeated a number of times for different values of T and κ.

The simulated annealing algorithm is initialized with pseudo-random image estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$, $\tilde{f}_0$ and $\tilde{h}_0$. Both positivity and support constraints are applied. The convolution $g(\mathbf{x})$ is scaled such that the energy of $g(\mathbf{x})$ is equal to the energy of $\tilde{f}_0 \odot \tilde{h}_0$.

Care must be taken when selecting the value of the temperature T, the rate of decreasing T for different cycle and the scale of perturbations κ, in order to ensure that the algorithm produces a global minimum $E[g]$ rather than being trapped in a local minimum.

McCallum (1990b) (*cf.* McCallum, 1990a) has demonstrated the capability of the algorithm to deconvolve real and positive images contaminated up to a high level of SNR = 20dB. The algorithm produced steadily decreasing error metric $E[g]$ as iterations proceeded and was well behaved. McCallum (1990b) mentions the disadvantage of the algorithm is its large computational cost.

3.4.2 Conjugate gradient method

The conjugate gradient methods (Press *et al.*, 1992, §10.5 and §10.6) have been devised to search for the minimum of a function E along a better or effective set of directions compared with methods such as the steepest descent method (Polak, 1971, §2.1).

The basic idea of the conjugate gradient methods is now described. Suppose the

given function E can be roughly approximated as a quadratic form, its approximate expression by Taylor series is

$$\begin{aligned} E(\mathbf{x}) &= E(\mathbf{p}) + \sum_m \frac{\partial E}{\partial x_m} x_m \Big|_{\mathbf{p}} + \frac{1}{2} \sum_{m,n} \frac{\partial^2 E}{\partial x_m \partial x_n} x_m x_n \Big|_{\mathbf{p}} + \ldots \\ &\approx c - \mathbf{b} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x}, \end{aligned} \tag{3.24}$$

where $\mathbf{p}$ is a particular point within a multidimensional coordinate system with coordinates $\mathbf{x} = \{x_m\}$ and

$$c = E(\mathbf{p}) \quad b_m = -\frac{\partial E}{\partial x_m}\Big|_{\mathbf{p}} \qquad A_{m,n} = \frac{\partial^2 E}{\partial x_m \partial x_n}\Big|_{\mathbf{p}}. \tag{3.25}$$

Taking the gradient of E using the approximation (3.24) gives

$$\nabla E = \mathbf{A} \cdot \mathbf{x} - \mathbf{b}. \tag{3.26}$$

At a given point $\mathbf{p}_{i-1}$, the function $E(\mathbf{p}_i)$ is minimized along the line $\mathbf{d}_i$. The next direction $\mathbf{d}_{i+1}$ is searched in a manner that the change in the gradient stays perpendicular to the previous direction $\mathbf{d}_i$, so as not to spoil the minimization along $\mathbf{d}_i$. That is

$$0 = \mathbf{d}_i \cdot \delta(\nabla E) = \mathbf{d}_i \cdot \mathbf{A} \cdot \mathbf{d}_{i+1}. \tag{3.27}$$

Once the two vectors $\mathbf{d}_i$ and $\mathbf{d}_{i+1}$ satisfy (3.27), they are said to be *conjugate*. If the relation holds pairwise for all members of the set of direction vectors $\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_I$, they are said to be a conjugate set. The conjugate gradient minimization method performs successive minimization of E along a conjugate set of directions. The procedure is continued until E stops decreasing. This method can avoid interminable cycling through the set of directions.

The application of conjugate gradient minimization methods to the phase retrieval problem has been considered by Nieto-Vesperinas (1986), Sasaki and Yamagami (1987) and Lane (1991). Its application to the blind deconvolution problem was proposed by Lane (1992).

The Conjugate Gradient Algorithm (Lane, 1992) is constituted by using a conjugate gradient routine to minimize a composite error metric E_c. E_c is formed by calculating how much the estimates at each iteration violating both the Fourier space constraints and the image space constraints.

$$E_c = E_{sp} + E_{im}, \tag{3.28}$$

where

$$E_{sp} = \int_{\Theta^{sp}} |G(\mathbf{u}) - \tilde{F}(\mathbf{u})\tilde{H}(\mathbf{u})|^2 d\mathbf{u}, \tag{3.29}$$

and

$$E_{im} = \int_{\Omega_i^f} |\tilde{f}(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x} + \int_{\Omega_i^h} |\tilde{h}(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x}, \tag{3.30}$$

where Θ^{sp} represents the whole spectral domain, and Ω_i^f and Ω_i^h have been defined in §3.2.2. E_{sp} measures by how much the spectral estimate of the convolution deviates from the given $G(\mathbf{u})$, and E_{im} represents the energy associated with the pixels of $\tilde{f}(\mathbf{x})$ and $\tilde{h}(\mathbf{x})$ which violate the image space constraints, such as the support and/or positivity. The constraints are thus incorporated into the error metric and the problem is reformulated as an unconstrained minimization problem (Fletcher, 1987) in the algorithm.

To implement conjugate gradient minimization, the gradients of the error metric E_c with respect to the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$, $\partial E_c/\partial \tilde{f}(\mathbf{x})$ and $\partial E_c/\partial \tilde{h}(\mathbf{x})$, respectively, must be calculated (Lane, 1992) and then used in a conjugate gradient routine E04DGF in the NAG library (1990).

Lane (1992) reported that the Conjugate Gradient Algorithm is a robust minimization routine with well-defined minimization problem (Fletcher, 1987; Gill and W. Murray, 1974). It incorporates the composite error metric E_c so that the estimates are generated with both of the image space error E_{im} and the Fourier space error E_{sp} being reduced and minimized. Restorations are obtained when the algorithm has found a local minimum E_c, i.e. when any perturbation of the estimate of $f(\mathbf{x})$ or $h(\mathbf{x})$ causes an increase in E_c.

Lane (1992) emphasised that there are local minima in the error metric for phase retrieval. As long as the local minima are close to the true solution, valuable restorations can result. For simple images that is often the case in practice.

Lane (1992) demonstrated using computer simulated data that the algorithm worked for simple images both uncontaminated and contaminated. The algorithm performed stably and produced a monotonic decrease in E_c. Lane (1992) pointed out that the technique is significantly less computationally intensive compared with comparative techniques based on simulated annealing (McCallum, 1990b). Lane (1992) stated that the algorithm is more robust than the Ayers and Dainty algorithm (*cf.* §3.2.2) and the Davey Algorithm (*cf.* §3.2.3). He pointed out two major problems of the Ayers and Dainty algorithm. One is that in the vicinity of the zeros of $H(\mathbf{u})$, small errors in $G(\mathbf{u})$ can result in large errors in $F(\mathbf{u})$. The other is that the iterative loop of the Ayers and Dainty algorithm cannot be equated with a steepest-descent search on an error metric and it thus lacks stable convergence properties.

The Conjugate Gradient Algorithm is implemented for comparison with an iterative algorithm in chapter 6.

3.4.3 Maximum likelihood method

The maximum likelihood method can be traced as early as the 1920's when it was employed by statistician Fisher (1922). The technique has been used to solve deconvolution

problems encountered in areas, such as in seismic data (Mendel, 1985) and communication systems. In the context of image processing, it is assumed that each pixel or sample of the image is an independent random variable. The method is based on the idea that different probability models generate different samples of the image. For given samples of a measurement $g(\mathbf{x})$, $g(\mathbf{x})$ is more likely to have come from some probability models than from others.

In order to apply the maximum likelihood method to solve the deconvolution problem, the following steps are involved (Mendel, 1990):

(i) Specify a probability model for the measurement $g(\mathbf{x})$, which includes the models for the true image $f(\mathbf{x})$, psf $h(\mathbf{x})$ and contamination $c(\mathbf{x})$, e.g. Gaussian model, autoregressive model, etc;

(ii) Establish some form of the likelihood function including the parameters concerned, e.g. the measurement, the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$, $\tilde{f}(\mathbf{x})$ and $\tilde{h}(\mathbf{x})$;

(iii) Maximize the likelihood function with respect to all of the unknown parameters and find the solution.

Likelihood is proportional to probability. The likelihood function is characterized by the probability density function. In probability, all of parameters in the probability model, such as mean and variance, are fixed and the measurements are then generated. In likelihood, the mean and variance are inferred from the given measurements, and the former chosen which are most likely generate the latter.

In practice it is often difficult to find the maximum likelihood solution from the given measurement directly, so numerical optimization techniques need to be employed. Gradient based methods (Lagendijk *et al.*, 1988), recursive methods (Tekalp *et al.*, 1986), penalized techniques (Synder and Miller, 1985; Roysam *et al.*, 1988) and expectation maximization algorithm (Shepp and Vardi, 1982; Lange *et al.*, 1987; Lewitt and Muehllehner, 1986; Holmes and Liu, 1989) have all been applied to obtain the maximum likelihood estimate.

Much work has been carried out on the application of the maximum likelihood method in blind deconvolution (Tekalp and Kaufman, 1988; Lagendijk *et al.*, 1990; Nakajima, 1993). Holmes (1992) applies the maximum likelihood method to solve the blind deconvolution problem of quantum-limited incoherent imagery, including fluorescence microscope imagery. Holmes (1992) specifies $f(\mathbf{x})$ to have the fluorescent probe distribution or the intensity function of an inhomogeneous Poisson random-point process, representing the positions of photon detection under the ideal condition. The psf of the optical system $h(\mathbf{x})$ owing to diffraction is also an inhomogeneous Poisson random-point process according to the constructive derivation of the Poisson process given in (Synder, 1975, p. 117). For the same reason the convolution of $f(\mathbf{x})$ and $h(\mathbf{x})$ is

an intensity function of a Poisson random-point process. Holmes (1992) employs the expectation maximization algorithm, formalized by Dempster *et al.* (1977), for a given number of iterations to find the solution of maximum likelihood estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ from the given measurement $g(\mathbf{x})$.

Holmes (1992) points out that positivity of the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ is preserved in his algorithm if the initial estimates are positive. The algorithm has the advantages of using a fundamentally sound optimization method and being a tractable and implementable algorithm. However, the limitations of the apparent divergence and the noise artifact are addressed.

Holmes (1992) demonstrates examples of implementing the algorithm and showed that his approach is workable to the blind deconvolution problem.

Fish *et al* (1995) recently propose a blind deconvolution algorithm similar to the Holmes algorithm. It is based on the well-known Richardson-Lucy Algorithm (Richardson, 1972; Lucy, 1974) which, given a recorded image $g(\mathbf{x})$ and an estimate of the psf $\tilde{h}$, produces the restoration

$$\tilde{f}^{j+1}(\mathbf{x}) = \left(\frac{g(\mathbf{x})}{\tilde{f}^{j}(\mathbf{x}) \odot \tilde{h}(\mathbf{x})} \odot \tilde{h}(-\mathbf{x}) \right) \tilde{f}^{j}(\mathbf{x}) \tag{3.31}$$

at the j^{th} iteration. Equation (3.31) is in a similar form to that used in the Holmes algorithm and, of course, is a solution of the maximum likelihood approach. A starting estimate of the image $\tilde{f}^{0}(\mathbf{x})$ is required to perform the Richardson-Lucy Algorithm which has become popular for standard deconvolution problems. Fish *et al* (1995) extend the Richardson-Lucy algorithm to restore the image and psf blindly and simultaneously. At i^{th} blind iteration, the estimate $\tilde{h}_i(\mathbf{x})$ is generated by performing the Richardson-Lucy algorithm in (3.31) with the roles of $\tilde{f}(\mathbf{x})$ and $\tilde{h}(\mathbf{x})$ interchanged for a specified number J iterations. The estimate of $h(\mathbf{x})$ at j^{th} Richardson-Lucy iteration of i^{th} blind iteration is produced by

$$\tilde{h}_i^{j+1}(\mathbf{x}) = \left(\frac{g(\mathbf{x})}{\tilde{h}_i^{j}(\mathbf{x}) \odot \tilde{f}_{i-1}(\mathbf{x})} \odot \tilde{f}_{i-1}(-\mathbf{x}) \right) \tilde{h}_i^{j}(\mathbf{x}), \tag{3.32}$$

$\tilde{h}_i(\mathbf{x})$ $(= \tilde{h}_i^{J}(\mathbf{x}))$ is then used to yield the new estimate $\tilde{f}_i(\mathbf{x})$ by running a further J iterations of the Richardson-Lucy Algorithm with

$$\tilde{f}_i^{j+1}(\mathbf{x}) = \left(\frac{g(\mathbf{x})}{\tilde{f}_i^{j}(\mathbf{x}) \odot \tilde{h}_i(\mathbf{x})} \odot \tilde{h}_i(-\mathbf{x}) \right) \tilde{f}_i^{j}(\mathbf{x}). \tag{3.33}$$

The difference between the algorithms of Holmes and Fish *et al* is that the latter runs several Richardson-Lucy iterations to generate each of the estimate of $f(\mathbf{x})$ and $h(\mathbf{x})$ at one blind iteration, while in the Holmes algorithm, only one maximum likelihood estimation is performed each iteration. Fish *et al.* (1995) claim that their results are superior.

The Fish *et al* algorithm preserves the properties of positivity and conservation of energy if the starting estimates are positive. Fish *et al* (1995) demonstrate the performance of their algorithm to deconvolve contaminated blurred image $g(\mathbf{x})$ using a simple binary image $f(\mathbf{x})$ featuring "cross" and a Gaussian psf $h(\mathbf{x})$. They also report that their algorithm was more tolerant of contamination and performed more stably compared with the Davey Algorithm (*cf.* §3.2.3). However it is clear that the latter was applied without a support constraint which is clearly detrimental to its performance (*cf.* §4.1.2).

3.4.4 Maximum entropy method

Unlike the minimization methods introduced in the previous sections, the maximum entropy method belongs to the category of regularization. The regularization method approaches the image restoration problem in the manner that any image estimate which is consistent with the given measurement is accepted as a feasible image. The restoration is then selected from the set of feasible image estimates according to some criterion specified by the regularization function.

Burch *et al* (1983) proposed a measure to determine the feasible images by setting the discrepancy between the given measured data $g(\mathbf{x})$ and the image estimate $\tilde{g}(\mathbf{x})$ as the chi-squared statistic $\chi^2[\tilde{g}(\mathbf{x})]$,

$$\chi^2[\tilde{g}(\mathbf{x})] = \int_{\mathbf{x}} \frac{(\tilde{g}(\mathbf{x}) - g(\mathbf{x}))^2 \, \mathrm{d}\mathbf{x}}{\sigma(\mathbf{x})^2} \, \mathrm{d}\mathbf{x}, \tag{3.34}$$

where $\sigma(\mathbf{x})$ is the standard deviation of the contamination. Burch *et al.* (1983) defined the set of feasible images $\tilde{g}(\mathbf{x})$ also satisfying

$$\chi^2[\tilde{g}(\mathbf{x})] = N^2, \tag{3.35}$$

where N^2 is the total number of pixels of the image $g(\mathbf{x})$.

The manner to determine the restoration is to optimize the regularization function associated with the set of feasible images. A common regularization function is entropy and the associate algorithm, the maximum entropy method. The entropy of $\tilde{g}(\mathbf{x})$, denoted by $S[\tilde{g}(\mathbf{x})]$, may be computed as

$$S[\tilde{g}(\mathbf{x})] = \int_{\mathbf{x}} \tilde{g}(\mathbf{x}) \log \tilde{g}(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \tag{3.36}$$

e.g. by Hildebrandt (1987), or as

$$S[\tilde{g}(\mathbf{x})] = -\int_{\mathbf{x}} \log \tilde{g}(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \tag{3.37}$$

as used by Ables (1974).

The application of the maximum entropy method to the blind deconvolution problem has been outlined by Gull and Skilling (1984). They recover the estimates of the

image $f(\mathbf{x})$ and psf $h(\mathbf{x})$ by maximizing a joint entropy of $f(\mathbf{x})$ and $h(\mathbf{x})$ subject to the constraint that the convolution estimate agrees with the given measurement determined by the chi-squared statistic in (3.34). The joint entropy is defined by

$$S[\tilde{f}(\mathbf{x}), \tilde{h}(\mathbf{x})] = \theta S[\tilde{f}(\mathbf{x})] + (1-\theta) S[\tilde{h}(\mathbf{x})], \quad 0 \leq \theta \leq 1, \tag{3.38}$$

where $S[\tilde{f}(\mathbf{x})]$ and $S[\tilde{h}(\mathbf{x})]$ represent the entropy of $\tilde{f}(\mathbf{x})$ and $\tilde{h}(\mathbf{x})$, separately, and are as defined in (3.36). θ, a constant located between 0 and 1, determines the relative contribution of psf and the image entropies. If θ is close to one, the image entropy will dominate and little smoothing of psf will occur, and *vice versa* when θ is close to zero. Gull and Skilling (1984) demonstrated the effectiveness of their algorithm to blindly deconvolve a binary image blurred by a L-shaped psf.

Hildebrandt (1987) and Newman and Hildebrandt (1987) discussed the problems associated with the Gull and Skilling algorithm, and modified the algorithm to achieve the better performance of the algorithm.

Chapter 4

FURTHER INVESTIGATION OF THE DAVEY ALGORITHM

The emphasis of this chapter is on further investigation of several aspects of the Davey Algorithm (§3.2.3). The review of blind deconvolution methods in chapter 3 has shown that the Davey Algorithm is superior to many of the other methods. It has advantages in being both computationally inexpensive to perform and capable of deconvolving substantially contaminated images. However, successful performance of this algorithm relies on suitable choices of values of certain initial parameters. These values are typically determined by trial-and-error, which is a tedious process and limits the routine application of the algorithm. Investigation and analysis regarding the effect of these parameters on the convergence of the algorithm are therefore the subject of §4.1.

The Wiener-like filter in (3.19) is a key part of the Fourier space constraint (*cf.* (3.20)) of the Davey Algorithm. The correctness of the form of this filter, however, has been challenged by Seldin and Fienup (1990). Since the filter has similar form to the standard Wiener filter (*cf.* (3.9)) it has been assumed to have identical properties to the latter. Such expectation is in conflict with some observations of the behaviour of the algorithm (Davey, 1989, §7.4.3) and therefore restricts further improvement. Extensive study of the Fourier space filter is presented in §4.2. The work presented in §4.1 and§4.2 provides both experimental and analytical results which lead to the extension to the Davey Algorithm described in chapter 5.

Thorough evaluation of the performance of the algorithm is by no means a trivial task. To this point evaluation of the Davey Algorithm's performance has been based on the true error $E_T[f_i]$ described in (3.21), which correlates reasonably well with human visual perception in most cases (as shown in the examples presented in §4.1). $E_T[f_i]$ is certainly useful as a preliminary assessment of the performance of the algorithm. However, it is not applicable when the algorithm is used to deconvolve real-world images, where $f(\mathbf{x})$ is not available. To apply the algorithm routinely in any practical situation, and/or to incorporate any error-based automatic strategy into the algorithm, it is important to seek some objective error measurement to monitor the performance of the algorithm. Section 4.3 is devoted to investigating that how well some objective error measures correlate with the true error or the visual quality of the restorations when

the algorithm is applied on different situations. As a consequence, a new scheme using minimum of an objective error measure to indicate the best estimates of the supports of $f(\mathbf{x})$ and $h(\mathbf{x})$, which is one of the two key parameters of the algorithm, is proposed.

For brevity the term "algorithm" will refer to the Davey Algorithm throughout the rest of this chapter, except where otherwise specifically stated.

To avoid readers needing to constantly refer back to chapter 3, the iterative loop and the principal equations relating to the algorithm are repeated here. The iterative process loop of the Davey Algorithm, which is similar to Figure 3.1, is shown in Figure 4.1.

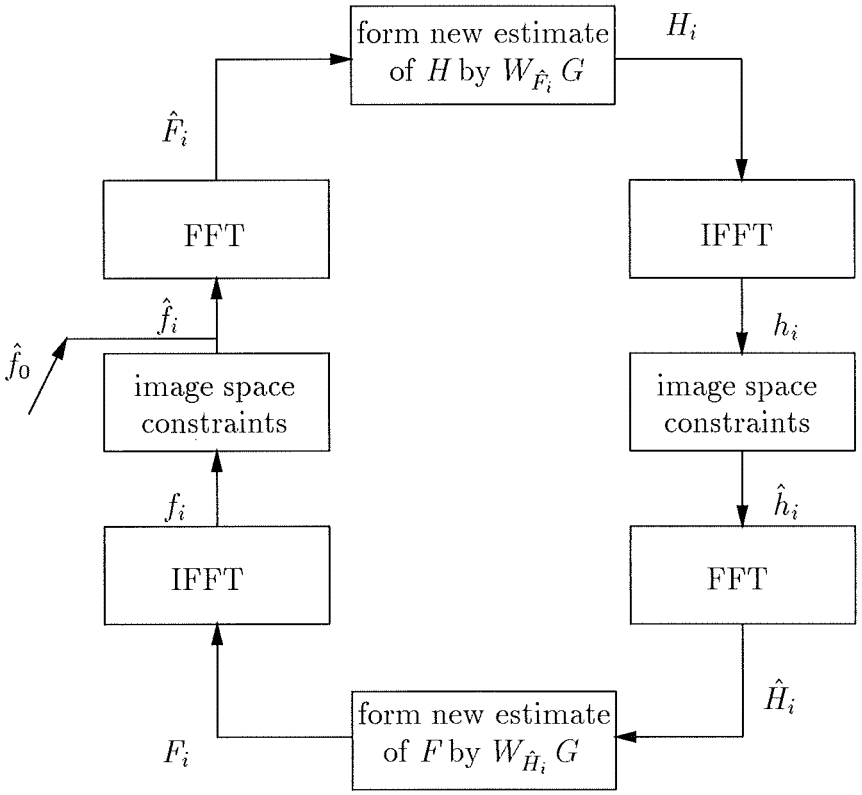


Figure 4.1 The iterative loop depicting the Davey Algorithm.

Equations (3.17) to (3.20) describe the Wiener-like filters and the Fourier space constraints for generating new estimates of $F(\mathbf{u})$ and $H(\mathbf{u})$ are repeated here,

$$W_{\hat{F}_i}(\mathbf{u}) = \frac{\hat{F}_i^*(\mathbf{u})}{|\hat{F}_i(\mathbf{u})|^2 + \beta / \ |\hat{F}_i(\mathbf{u})|^n}, \tag{4.1}$$

$$H_i(\mathbf{u}) = W_{\hat{F}_i}(\mathbf{u})G(\mathbf{u}), \tag{4.2}$$

$$W_{\hat{H}_i}(\mathbf{u}) = \frac{\hat{H}_i^*(\mathbf{u})}{|\hat{H}_i(\mathbf{u})|^2 + \beta / \ |\hat{H}_i(\mathbf{u})|^n}, \tag{4.3}$$

and

$$F_i(\mathbf{u}) = W_{\hat{H}_i}(\mathbf{u})G(\mathbf{u}), \tag{4.4}$$

where β is the filter constant and n is the exponent which is chosen as two by Davey *et al* (1989).

4.1 FACTORS WHICH INFLUENCE THE CONVERGENCE OF THE ALGORITHM

There are many factors which influence the convergence of the algorithm, such as the filter constant β, the initial support estimates of the true image $f(\mathbf{x})$ and psf $h(\mathbf{x})$, the starting estimate of $f(\mathbf{x})$, the contamination level of the measurement and the sizes of $f(\mathbf{x})$ and $h(\mathbf{x})$. However, for a given measurement, successful performance of the algorithm is mainly affected by the choices of first two parameters: the value of β and the sizes of the support estimates. These effects are investigated and discussed in §4.1.1 and §4.1.2.

4.1.1 Effect of the filter constant β on convergence

The filter constant β, together with its denominator in the Wiener-like filter in (4.3), adds an extra item $\beta/|\hat{H}_i(\mathbf{u})|^{n+1}$ to the ideal deconvolution operator, the inverse filter (3.2). This extra item is designed to enable the filter to cope with the contamination $c(\mathbf{x})$ presented in the blurred image $g(\mathbf{x})$ and the bias which is necessarily introduced by the Wiener filter (Davey, 1989, §7.4.1.2). In real-world applications $c(\mathbf{x})$ is unknown *a priori*, consequently β is selected through trial-and-error. Experiments show that the choice of the value of β significantly affects the convergence of the algorithm. An unsuitable value of β may lead to a spurious estimate of $f(\mathbf{x})$.

To investigate the effect of β on the convergence of the algorithm, the algorithm was applied to a blurred image $g(\mathbf{x})$ (shown in Figure 4.2(c)) which was formed by the convolution of a bilevel 32×32-pixel image $f(\mathbf{x})$ (Figure 4.2(a)) and a disc psf $h(\mathbf{x})$ (Figure 4.2(b)). $g(\mathbf{x})$ was contaminated with Gaussian noise to SNR of 30dB. $f(\mathbf{x})$ was real and positive and $h(\mathbf{x})$ had uniform value within a circular region 7 pixels in diameter. This disc psf is an ideal model of photographic out-of-focus blur. Both $f(\mathbf{x})$ and $h(\mathbf{x})$ were zero packed to extent 64×64 pixels, i.e. $f(\mathbf{x})$ and $h(\mathbf{x})$ were centered in a larger arrays of 64×64 pixels which had zero values for all other pixels. The purpose of zero packing $f(\mathbf{x})$ and $h(\mathbf{x})$ is to ensure that all the information within $g(\mathbf{x})$ is preserved. The initial estimate of $f(\mathbf{x})$ (shown in Figure 4.2(d)), denoted $\hat{f}_0(\mathbf{x})$, was formed by pseudo-random numbers uniformly distributed between 0 and 1 within the support estimate of $f(\mathbf{x})$, which is called the pseudo-random image henceforth. The supports of $f(\mathbf{x})$ and $h(\mathbf{x})$ are assumed to be known precisely. The algorithm was repeatedly operated, each time using one value of β ranging from 5×10^{-6} to 5×10^{-10} and running for a maximum of 100 iterations. The iteration was stopped when the minimum true error of the image occurred. Restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$, i.e. $f_i(\mathbf{x})$ and $h_i(\mathbf{x})$, were recorded before the image space constraints were applied at this iteration. These are presented in Figure 4.3.

True error curves, $E_T[f_i]$, correspond to the restored image in Figure 4.3 are shown

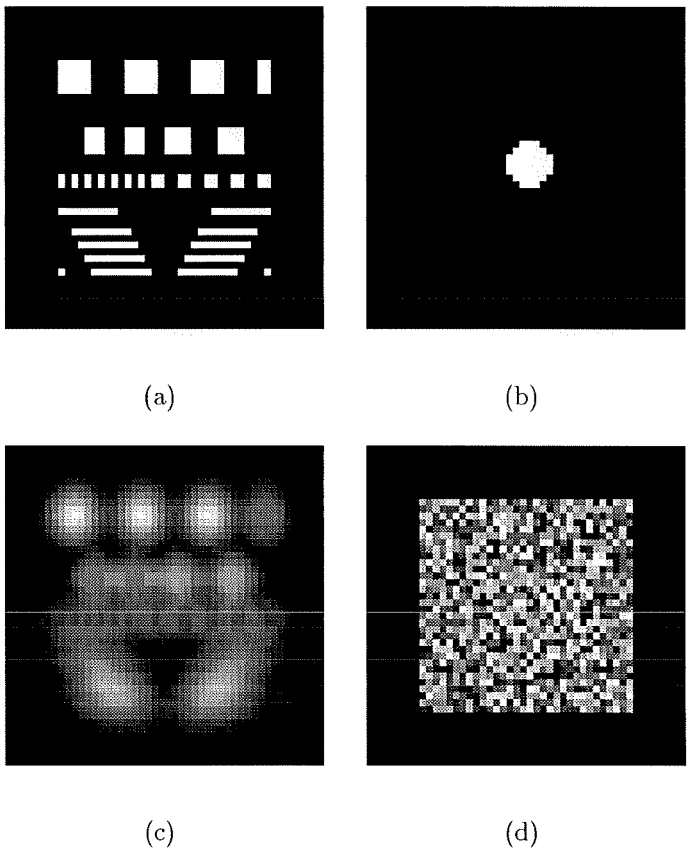


Figure 4.2 (a) Real positive true image $f(\mathbf{x})$; (b) the psf $h(\mathbf{x})$; (c) blurred image $g(\mathbf{x}) = f(\mathbf{x}) \odot h(\mathbf{x}) + c(\mathbf{x})$ with SNR = 30dB; and (d) starting image estimate $f_0(\mathbf{x})$. The central 48×48 pixels out of total size of 64×64 pixels are displayed.

in Figure 4.4. The effect of β on the convergence of the algorithm is clearly demonstrated. Note that a good choice of β (i.e. 5×10^{-8} in this example) produces very fast convergence.

The results presented in Figure 4.3 show that the performance of the algorithm is significantly affected by the value of β. When β was too small (i.e. $\beta = 5 \times 10^{-10}$), the restorations $f_i(\mathbf{x})$ and $h_i(\mathbf{x})$, as displayed in Figures 4.3(e) and 4.3(j), were swamped by the contamination; these restorations, especially $f_i(\mathbf{x})$, have little resemblance to their corresponding true images. In Figure 4.3(j), the estimate of $h(\mathbf{x})$ shows no evidence of a useful recovery of $h(\mathbf{x})$. This failure of the algorithm was due to inappropriate choices of the value of β and different rates of recovery of the two estimates. When the true error of $f_i(\mathbf{x})$ reached its minimum first, after several iterations, $h_i(\mathbf{x})$ was not even slightly recovered. Note when $h_i(\mathbf{x})$ was used to generate $f_i(\mathbf{x})$, only the part of $h_i(\mathbf{x})$ within the psf support was used. Therefore, the influence of the poor estimate of $h_i(\mathbf{x})$ on $f_i(\mathbf{x})$ in this case was not marked.

When β was large (i.e. $\beta = 5 \times 10^{-6}$), the restorations $f_i(\mathbf{x})$ and $h_i(\mathbf{x})$ were much less degraded by high frequency contamination. However, they were still quite blurred

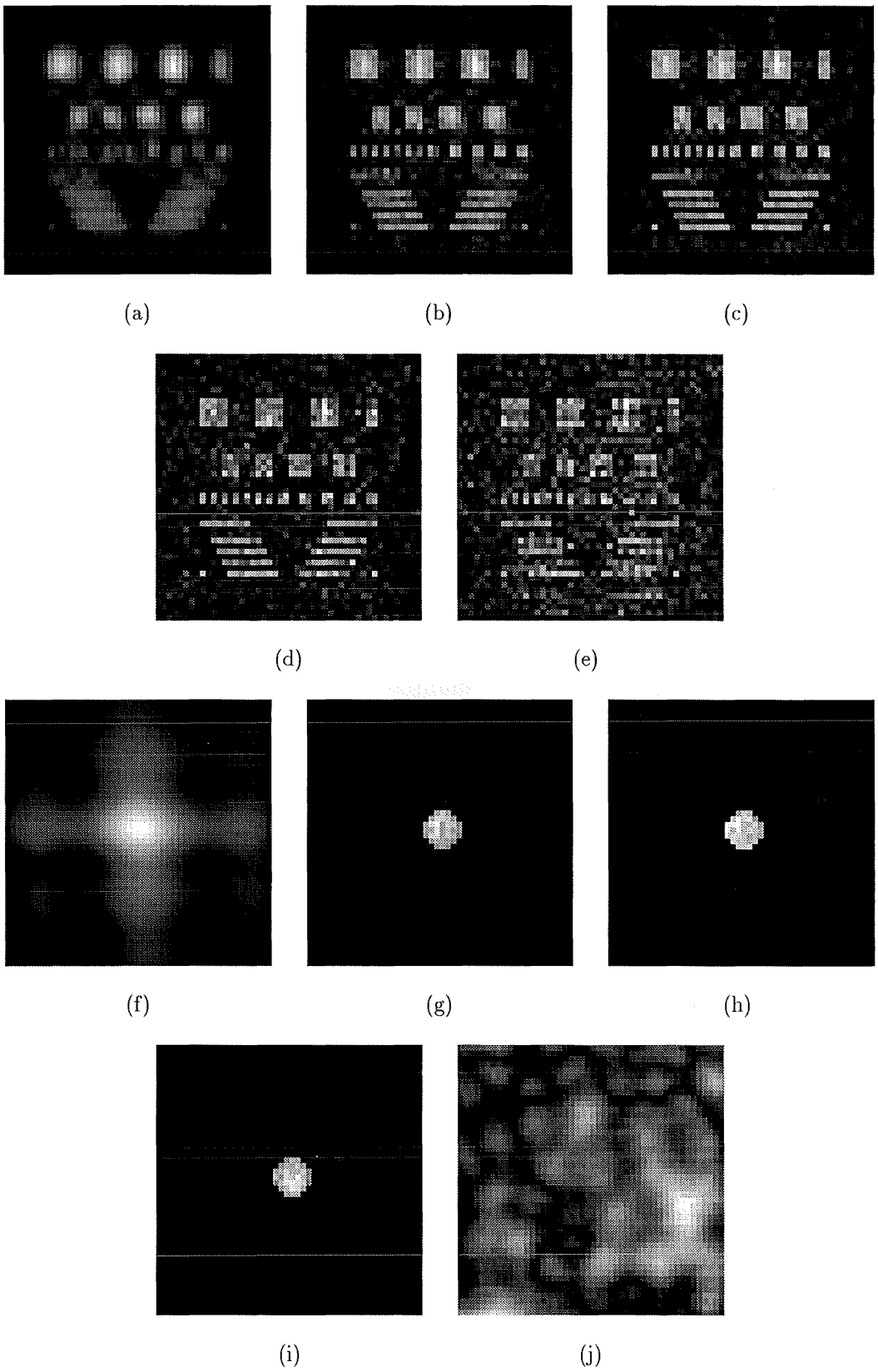


Figure 4.3 Demonstration of the effect of β on the performance of the algorithm, corresponding to the images shown in Figure 4.2. (a)-(e) restorations of $f(\mathbf{x})$ obtained using $\beta = 5 \times 10^{-6}, 5 \times 10^{-7}, 5 \times 10^{-8}, 5 \times 10^{-9}$ and 5×10^{-10}, respectively; (f)-(j) restorations of $h(\mathbf{x})$ corresponding to (a)-(e).

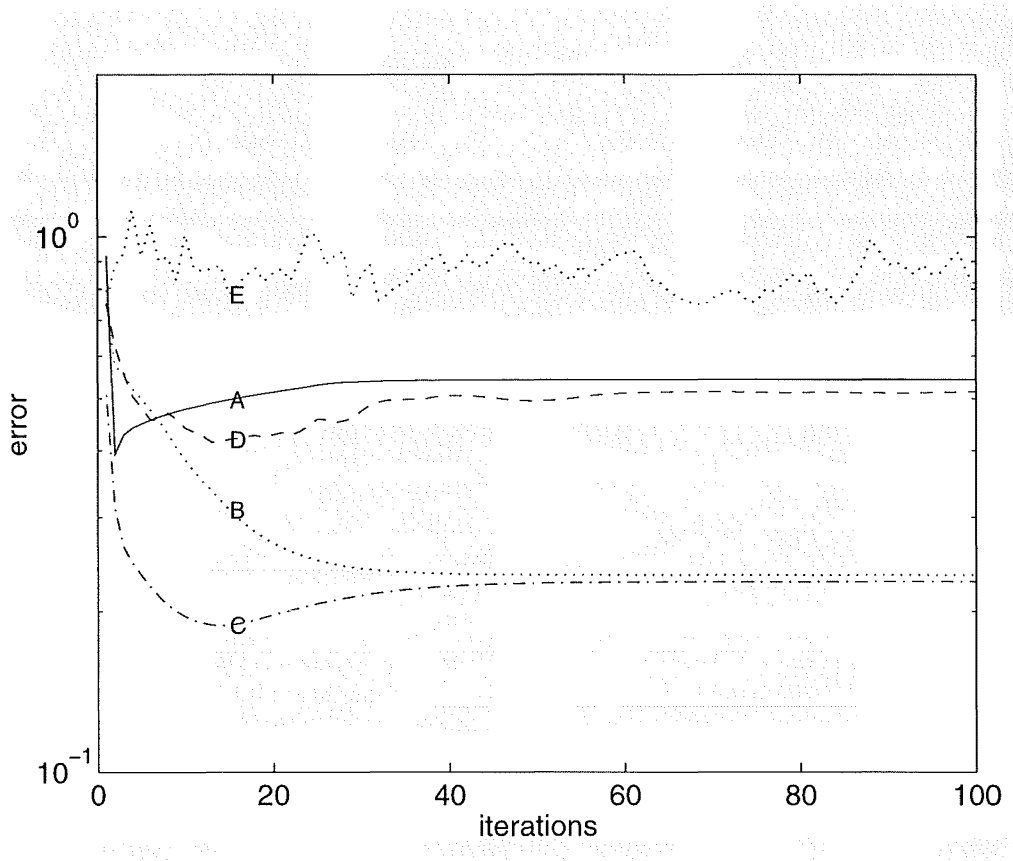


Figure 4.4 True error of $f_i(\mathbf{x})$ versus iterations, obtained when applying the Davey Algorithm to $g(\mathbf{x})$ in Figure 4.2(c) using different values of β: curves A - E corresponding to restorations in Figure 4.3 with $\beta = 5 \times 10^{-6}, 5 \times 10^{-7}, 5 \times 10^{-8}, 5 \times 10^{-9}$ and 5×10^{-10} respectively.

as illustrated in Figures 4.3(a) and 4.3(f). This blurredness suggests that significant information in the high spatial frequency range was lost. Consequently, there is no obvious evidence of successful deblurring having occurred for $f_i(\mathbf{x})$ and $h_i(\mathbf{x})$. For an appropriate choice of β, a good balance is achieved between deblurring and contamination amplification in $f_i(\mathbf{x})$. The restorations presented in Figures 4.3(c) and 4.3(h) display the details of $f(\mathbf{x})$ and $h(\mathbf{x})$ and demonstrate successful blind deconvolution.

The true errors $E_T[f_i]$ in Figure 4.4 show that $E_T[f_i]$ is lowest when β is most suitably chosen. When β is either too small or too large, the related $E_T[f_i]$ increases.

An explanation for the appreciable effect of the filter constant β on the convergence of the algorithm, or equivalently the quality of the restoration, is advanced in §4.2.2.

4.1.2 Effect of support on convergence

The essential difference between the algorithm investigated here and that of Ayers and Dainty (§3.2.2) is that the former applies a support constraint in image space and the latter does not. The support constraint is applied by windowing an image, say $q(\mathbf{x})$, by a mask $M_q(\mathbf{x})$. $M_q(\mathbf{x})$ is set to unity for the pixels within the support of $q(\mathbf{x})$, denoted $S[q(\mathbf{x})]$, and is set to zero otherwise. Since the supports for both the true image $f(\mathbf{x})$ and psf $h(\mathbf{x})$ are in most practical cases unknown (or at least imperfectly known) *a priori* and must be estimated from the given blurred image, $M_q(\mathbf{x})$ is set according to the estimate of the support $S[q(\mathbf{x})]$ and simply called support estimate in the rest of this thesis. Due to the fact that restorations using the image box(§2.2) as the support estimate (see Figures 4.5(b) and 4.5(e) in the following example) are comparable in quality with that using the exact support (see Figures 4.3(c) and 4.3(h)), the best support estimate $M_q(\mathbf{x})$ is therefore assumed to be the image box $B_q(\mathbf{x})$ and consequently $M_q(\mathbf{x})$ is chosen to be rectangular shape hereafter.

To investigate the effect of support on the convergence, the algorithm was, respectively, applied to two blurred images, both of which were real and positive. The first blurred image was the same as that shown in Figure 4.2(c), where $g(\mathbf{x})$ was contaminated to a level of SNR = 30dB. The second blurred image was generated by convolving a bilevel image $f(\mathbf{x})$ featuring a 'J' shape (shown in Figure 4.6(a)) with the psf $h(\mathbf{x})$ (shown in Figure 4.6(b)). $g(\mathbf{x})$ was contaminated to a level of SNR = 20dB as shown in Figure 4.6(c). The image $f(\mathbf{x})$ was of extent 16 pixels in each coordinate direction. The psf had a circular support of diameter 7 pixels. $f(\mathbf{x})$ and $h(\mathbf{x})$ were each embedded in a 32×32-pixel array.

The support estimate of $g(\mathbf{x})$, $M_g(\mathbf{x})$, is determined by equating it to the image box of $g(\mathbf{x})$, $B_g(\mathbf{x})$, which can be obtained by thresholding $g(\mathbf{x})$ appropriately (Davey, 1989, §7.4.1.1). Once either the support estimate of the image $M_f(\mathbf{x})$ or the support estimate of psf $M_h(\mathbf{x})$ is estimated, the other can be determined by knowledge of $M_g(\mathbf{x})$ and invocation of the convolution theorem. In the examples presented here, three cases of different combinations of the sizes of $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$ were considered.

case a) The extent of $M_f(\mathbf{x})$ in each coordinate direction is larger than the corresponding extent of $B_f(\mathbf{x})$ and the extent of $M_h(\mathbf{x})$ is smaller than that of $B_h(\mathbf{x})$. In other words, $M_f(\mathbf{x})$ is overestimated and $M_h(\mathbf{x})$ is underestimated;

case b) $M_f(\mathbf{x}) = B_f(\mathbf{x})$ and $M_h(\mathbf{x}) = B_h(\mathbf{x})$;

case c) $M_f(\mathbf{x})$ is underestimated and $M_h(\mathbf{x})$ is overestimated, i.e. $M_f(\mathbf{x}) < B_f(\mathbf{x})$ and $M_h(\mathbf{x}) > B_h(\mathbf{x})$.

In both examples, the algorithm started with pseudo-random image $\hat{f}_0(\mathbf{x})$ (shown in Figures 4.2(d) and 4.6(d)) and was applied repeatedly with different sizes of the support

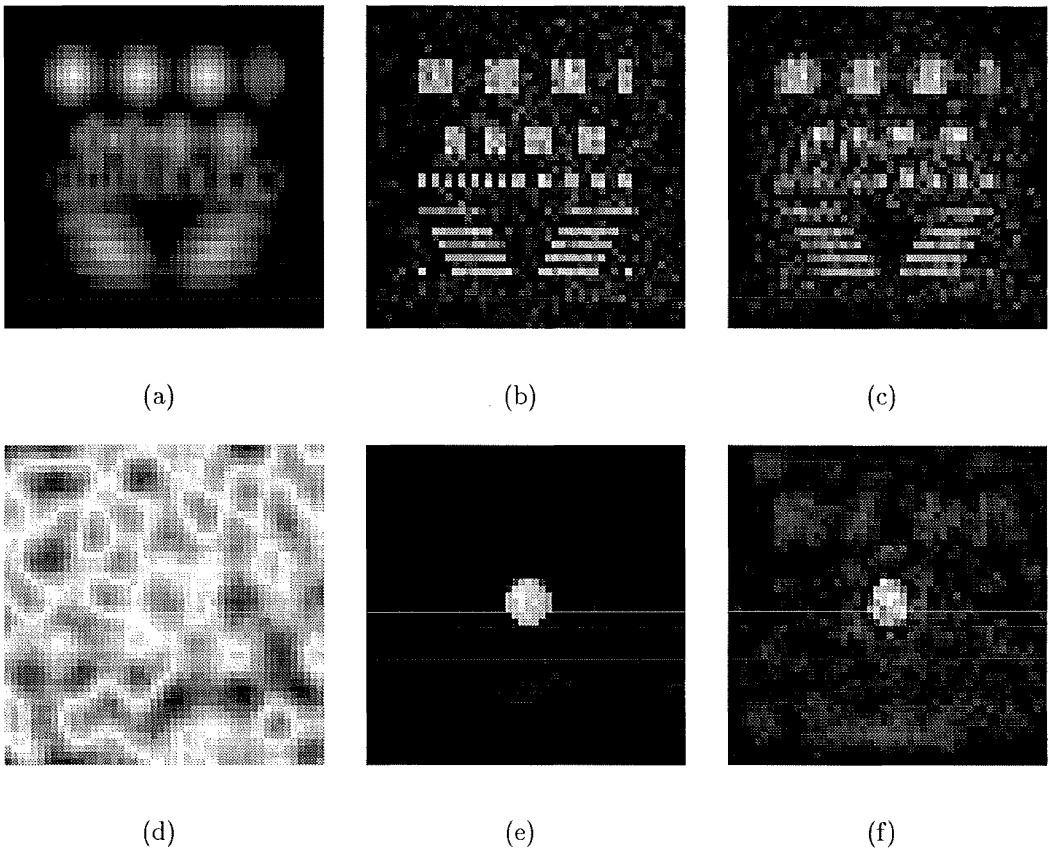


Figure 4.5 Demonstration of the effect of support on the convergence of the algorithm, $\beta = 5 \times 10^{-8}$, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$, corresponding to Figures 4.2(a) and 4.2(b), obtained using different sizes of support estimates: (a) and (d) case a), $M_f(\mathbf{x}) = 36 \times 36$ pixels, $M_h(\mathbf{x}) = 3 \times 3$ pixels; (b) and (e) case b), $M_f(\mathbf{x}) = 32 \times 32$ pixels, $M_h(\mathbf{x}) = 7 \times 7$ pixels; (c) and (f) case c), $M_f(\mathbf{x}) = 28 \times 28$ pixels, $M_h(\mathbf{x}) = 11 \times 11$ pixels.

estimates described in cases a) to c). An appropriate value of $\beta = 5 \times 10^{-8}$ was used for all cases. The resultant restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ are shown in Figure 4.5 for the first example and in Figure 4.6 for the second.

The true error curves $E_T[f_i]$ for the "blocks" example, corresponding to Figures 4.5(a), 4.5(b) and 4.5(c), are plotted in Figure 4.7. Figure 4.7 demonstrates that the performance of the algorithm is influenced by the initial support estimates, $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$, throughout the entire iterative process. Clearly, the best looking result (Figure 4.5(b)) coincides with the lowest error.

Experiments show when $M_f(\mathbf{x}) > B_f(\mathbf{x})$ and $M_h(\mathbf{x}) < B_h(\mathbf{x})$ (see case a)), the restorations $f_i(\mathbf{x})$ (see Figures 4.5(a), 4.6(e) and 4.6(f)) still appear blurred, while $h_i(\mathbf{x})$ (Figures 4.5(d) and 4.6(i)) appear severely truncated. The degree of the blurring on $f_i(\mathbf{x})$ varies with the severity of the overestimation of $M_f(\mathbf{x})$. When $M_f(\mathbf{x}) = B_f(\mathbf{x})$ and $M_h(\mathbf{x}) = B_h(\mathbf{x})$ (case b)), the restorations $f_i(\mathbf{x})$ and $h_i(\mathbf{x})$, illustrated in Figures 4.5(b) and 4.5(e), and Figures 4.6(g) and 4.6(k), show a good measure of agreement

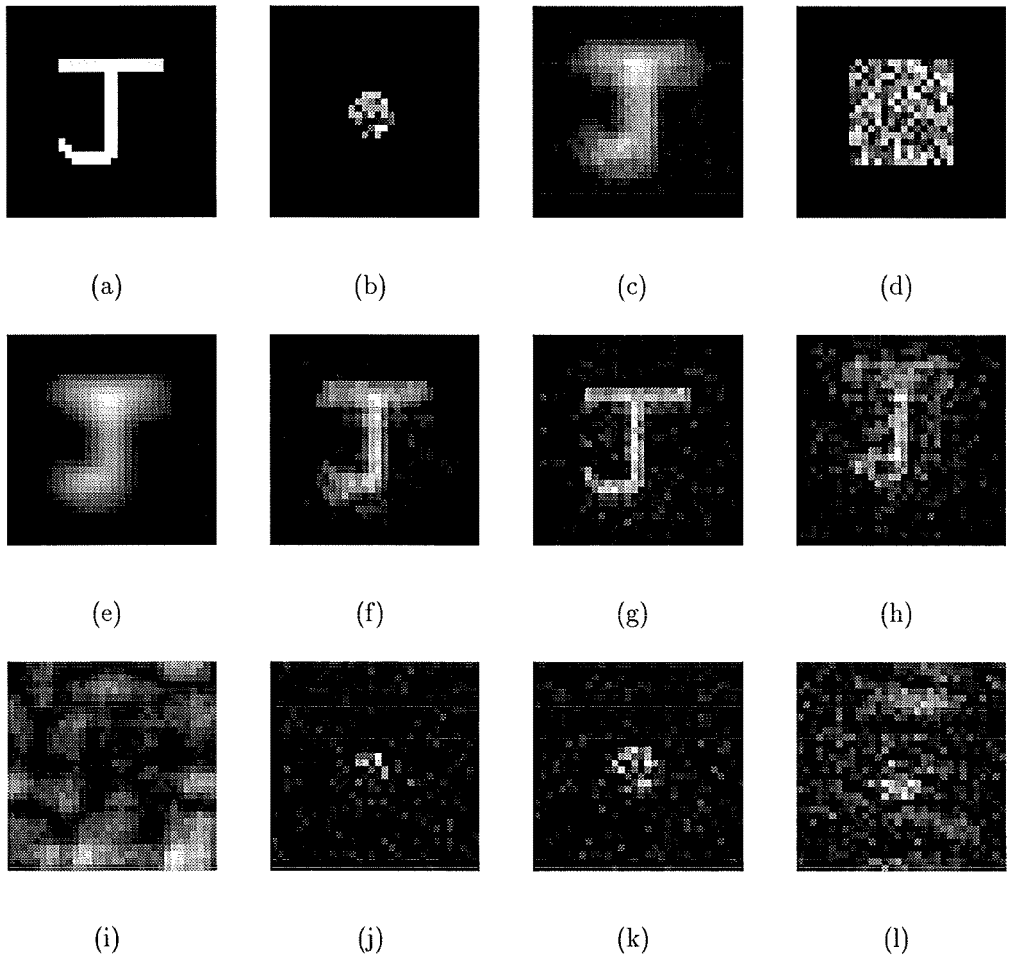


Figure 4.6 Second example of the effect of support on the convergence of the algorithm, $\beta = 10^{-6}$: (a) $f(\mathbf{x})$; (b) $h(\mathbf{x})$; (c) $g(\mathbf{x}) = f(\mathbf{x}) \odot h(\mathbf{x}) + c(\mathbf{x})$ with SNR = 20dB; (d) $\hat{f}_0(\mathbf{x})$. Restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ obtained using following sizes of support estimates: (e) and (i) case a) $M_f(\mathbf{x}) = 20 \times 20$ pixels, $M_h(\mathbf{x}) = 3 \times 3$ pixels; (f) and (j) case a) $M_f(\mathbf{x}) = 18 \times 18$ pixels, $M_h(\mathbf{x}) = 5 \times 5$ pixels; (g) and (k) case b) $M_f(\mathbf{x}) = 16 \times 16$ pixels, $M_h(\mathbf{x}) = 7 \times 7$ pixels; (h) and (l) case c) $M_f(\mathbf{x}) = 14 \times 14$ pixels, $M_h(\mathbf{x}) = 9 \times 9$ pixels.

with their true images and psfs. By comparison, when $M_f(\mathbf{x}) < B_f(\mathbf{x})$ and $M_h(\mathbf{x}) > B_h(\mathbf{x})$ (see case c)), the restorations are less sensitive to the inaccuracies of the support estimates $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$. This is demonstrated by the results presented in Figure 4.5. The restorations $f_i(\mathbf{x})$ (Figure 4.5(c)) and $h_i(\mathbf{x})$ (Figure 4.5(f)) reveal considerable detail. However when the relative extent of $B_f(\mathbf{x})$ at each direction compared with $B_h(\mathbf{x})$ is small, the effect of the underestimated $M_f(\mathbf{x})$ and overestimated $M_h(\mathbf{x})$ on the restorations is more severe. Considerable contamination occurs in the restorations as shown in Figures 4.6(h) and 4.6(l) although there is sign of deblurring in $f_i(\mathbf{x})$ and $h_i(\mathbf{x})$. The effect of the relative extent of $B_f(\mathbf{x})$ and $B_h(\mathbf{x})$ in case c) on the performance of the algorithm is further discussed in the latter part of the section. Clearly, a poor choice for the support estimates has a significant effect on the quality of the restoration.

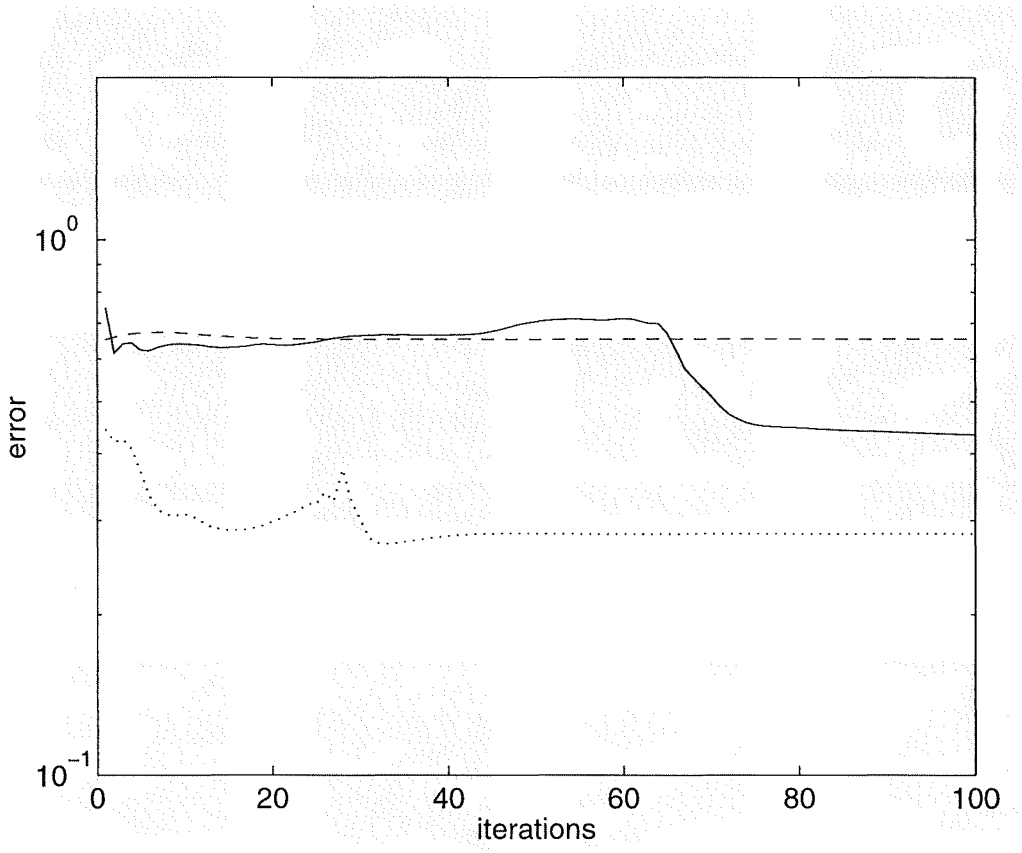


Figure 4.7 The true errors $E_T[f_i]$ obtained when the algorithm was applied on Figure 4.2(c) using different support estimates, dashed curve: case a), $M_f(\mathbf{x}) = 36 \times 36$ pixels, $M_h(\mathbf{x}) = 3 \times 3$ pixels; dotted curve: case b), $M_f(\mathbf{x}) = 32 \times 32$ pixels, $M_h(\mathbf{x}) = 7 \times 7$ pixels and continuous curve: case c), corresponds to $M_f(\mathbf{x}) = 28 \times 28$ pixels, $M_h(\mathbf{x}) = 11 \times 11$ pixels. These error curves correspond to restorations shown in Figure 4.5.

The reason for the effect of the support size on the convergence of the algorithm is now discussed. Applying a support constraint to an estimate is equivalent to multiplying the estimate by a window function. In Fourier space, this is equivalent to convolving the Fourier transforms of the image estimate and the window function. For case a), where $M_f(\mathbf{x}) > B_f(\mathbf{x})$ and $M_h(\mathbf{x}) < B_h(\mathbf{x})$, after applying support constraint with underestimated $M_h(\mathbf{x})$, the estimate $\hat{h}_i(\mathbf{x})$ is truncated. Since $h(\mathbf{x})$ has either uniform or pseudo-random values distributed over a circular region in the examples of Figures 4.5 and 4.6, the truncation of $\hat{h}_i(\mathbf{x})$ here is equivalent to spatial compression of $\hat{h}_i(\mathbf{x})$ in each direction, which corresponds to expansion of $\hat{H}_i(\mathbf{u})$ in the spatial frequency space. Local minima of $|\hat{H}_i(\mathbf{u})|$ are more widely spaced and, for a considerable range of higher frequencies, $|\hat{H}_i(\mathbf{u})|$ is affected and has larger values than the estimate of $|H(\mathbf{u})|$ obtained from the correct estimate $M_h(\mathbf{x})$. When $\hat{H}_i(\mathbf{u})$ is used to generate a new estimate $F_i(\mathbf{u})$, the larger values of $|\hat{H}_i(\mathbf{u})|$ attenuate $|F_i(\mathbf{u})|$ at the related frequencies and result in the loss of certain details of the image. The restoration $f_i(\mathbf{x})$ is therefore still a blurred

version of its true version as shown in Figures 4.5(a), 4.6(e) and 4.6(f).

There is a second consideration. In the majority of image blurring situations, the size of the image is considerably larger than that of the psf. That is $B_f(\mathbf{x}) > B_h(\mathbf{x})$. Assuming $B_g(\mathbf{x})$ can be correctly estimated from $g(\mathbf{x})$, the number of pixels overestimated for the extent of $M_f(\mathbf{x})$ in one direction equals that underestimated for $M_h(\mathbf{x})$ at the same direction, and *vice versa*. When $B_f(\mathbf{x}) > B_h(\mathbf{x})$, the relative errors of the support estimates $M_f(\mathbf{x})$, denoted $E[M_f]$, is smaller than $E[M_h]$ whenever the support estimates are in error. $E[M_f]$ and $E[M_h]$ are defined by

$$E[M_f] = \left| \frac{area[M_f] - area[B_f]}{area[B_f]} \right| \tag{4.5}$$

and

$$E[M_h] = \left| \frac{area[M_h] - area[B_h]}{area[B_h]} \right|, \tag{4.6}$$

where $area(M_f)$ and $area(B_f)$ represent the non-zero areas of $M_f(\mathbf{x})$ and $B_f(\mathbf{x})$ respectively. For the example of Figures 4.5 and 4.5(d), $B_f(\mathbf{x})$ (32×32 pixels) is greater than $B_h(\mathbf{x})$ (7×7 pixels) and $E[M_f]$ (27%) is smaller than $E[M_h]$ (82%). However, in this case, the support estimate with smaller error, $M_f(\mathbf{x})$, does not give much more influence to the recovering process than $M_h(\mathbf{x})$ does. Large amount of information of $h(\mathbf{x})$ could possibly be retrieved from $\hat{f}_i(\mathbf{x})$ and $g(\mathbf{x})$. However, when the estimate of $h(\mathbf{x})$, $\hat{h}_i(\mathbf{x})$, is alternatively used to produce $f_i(\mathbf{x})$ at each iteration, $\hat{h}_i(\mathbf{x})$ has been truncated by the underestimated $M_h(\mathbf{x})$ due to the application of the support constraint. A substantial amount of the useful information in $h(\mathbf{x})$ is thus discarded. The consequent new estimate of $f(\mathbf{x})$ is considerably affected by the larger error of $M_h(\mathbf{x})$. Such influence carries on and impacts the performance of the algorithm. Note the failure of the algorithm on recovering $h(\mathbf{x})$ shown in Figure 4.5(d) is due to wrong estimation of the supports and much later occurrence of the minimum true error for $h(\mathbf{x})$ compared with that for $f(\mathbf{x})$. The restored psf could be much better if it is recorded at the iteration where $E_T[h_i]$ is minimum.

Case b) When $M_f(\mathbf{x}) = B_f(\mathbf{x})$ and $M_h(\mathbf{x}) = B_h(\mathbf{x})$, the support constraint consistently supplies correct information about the size and the location that $f(\mathbf{x})$ and $h(\mathbf{x})$ occupy in their respective estimates at each iteration. Consequently, the support constraint assists the deconvolution process to converge towards the solution.

Case c) When $M_f(\mathbf{x}) < B_f(\mathbf{x})$ and $M_h(\mathbf{x}) > B_h(\mathbf{x})$, the underestimated $M_f(\mathbf{x})$ truncates $\hat{f}_i(\mathbf{x})$. However, overestimating $M_h(\mathbf{x})$ does not necessarily cause the actual extents of the estimate $h_i(\mathbf{x})$ to be as large as that of $M_h(\mathbf{x})$ and larger error of $M_h(\mathbf{x})$ does not necessarily have more influence to the performance of the algorithm than the smaller error of $M_f(\mathbf{x})$ does. Consider the example presented in Figures 4.5(c) and 4.5(f), at the early iterations of the process, the basic structure of $h_i(\mathbf{x})$ has not formed yet, $h_i(\mathbf{x})$ still appears as a random-like estimate. At this stage enforcing a support constraint using overestimated $M_h(\mathbf{x})$ enlarges the extents of $\hat{h}_i(\mathbf{x})$ the same

as that of $M_h(\mathbf{x})$ (here $E[M_h] = 147\%$). In spatial frequency space, that corresponds to a contraction of $\hat{H}_i(\mathbf{u})$. Local minima of $|\hat{H}_i(\mathbf{u})|$ are more narrowly spaced and for a substantial range of higher frequencies, $|\hat{H}_i(\mathbf{u})|$ has much smaller values than those which would result from the correct estimate $M_h(\mathbf{x})$. The values of the consequent new estimate $|F_i(\mathbf{u})|$ at the related frequencies, which often appears as contamination in $|F_i(\mathbf{u})|$, are therefore overemphasized and more severe contamination in $F_i(\mathbf{u})$ is produced. This is only the influence of the overestimated $M_h(\mathbf{x})$ on the estimates in the early iterations. Consider on the other hand the effect of the underestimated $M_f(\mathbf{x})$ on the performance of the algorithm. Since the error of $M_f(\mathbf{x})$ is only 23%, the remaining 77% information of the support of $f(\mathbf{x})$ as well as other useful constraints can still be used to retrieve considerable amount of information of $h(\mathbf{x})$. As iterations proceed, the restoration $h_i(\mathbf{x})$ is gradually recovered as shown in Figure 4.5(f). $h_i(\mathbf{x})$ is of similar extent in each coordinate direction to that of $S_h(\mathbf{x})$ rather than $M_h(\mathbf{x})$. At this stage, the effect of overestimated $M_h(\mathbf{x})$ in support constraint is to maintain values of two parts of $h_i(\mathbf{x})$ positioned within $M_h(\mathbf{x})$ and produce $\hat{h}_i(\mathbf{x})$: one is the key part of $h_i(\mathbf{x})$ within $S_h(\mathbf{x})$ and the other is the noisy background of $h_i(\mathbf{x})$ located between $S_h(\mathbf{x})$ and $M_h(\mathbf{x})$. In Fourier space, the spectral estimate $\hat{H}_i(\mathbf{u})$ is formed by addition of two partial spectra of $\hat{h}_i(\mathbf{x})$: one corresponds to the former part of $\hat{h}_i(\mathbf{x})$ and contains most of the useful information of $\hat{H}_i(\mathbf{u})$; the other relates to the latter part of $\hat{h}_i(\mathbf{x})$ and involves most of the noisy part of $\hat{H}_i(\mathbf{u})$ due to the noisy background of $\hat{h}_i(\mathbf{x})$ windowed by the overestimated $M_h(\mathbf{x})$. Since the values of the noisy background are significantly smaller than that of the main features, the magnitudes of its spectrum are significantly smaller too. This part of spectrum serves as contamination and is insignificant. When the restored $\hat{h}_i(\mathbf{x})$ is alternatively involved to deconvolve $f_i(\mathbf{x})$, $f_i(\mathbf{x})$ is consequently improved. There is no obvious sign of truncation in $f_i(\mathbf{x})$, some information of $f(\mathbf{x})$ beyond the underestimated $M_f(\mathbf{x})$ is also retrieved as shown in Figure 4.5(c). The performance of the algorithm is therefore significantly affected by the smaller error of $M_f(\mathbf{x})$ in case c).

It is noted that when $M_f(\mathbf{x}) < B_f(\mathbf{x})$ and $M_h(\mathbf{x}) > B_h(\mathbf{x})$, the relative size of $B_f(\mathbf{x})$ compared with $B_h(\mathbf{x})$ has significant effect on the restorations. This is due to the fact that for a fixed size of $B_h(\mathbf{x})$, if its estimate $M_h(\mathbf{x})$ is underestimated by the same number of pixels as $M_f(\mathbf{x})$ is overestimated, the smaller the relative size of $B_f(\mathbf{x})$ compared to $B_h(\mathbf{x})$, the larger the error of $M_f(\mathbf{x})$, and consequently the worse the restorations. In the example presented in Figure 4.5, the relative size of $B_f(\mathbf{x})$ compared with $B_h(\mathbf{x})$ is 32×32/7×7. The image and psf are of more comparable size in the example shown in Figure 4.6 where the ratio of extents is 16×16/7×7. The latter ratio of extents is 4 times smaller than the former. Consequently, compared with each pair of restorations obtained from applying correct support estimates, $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$, the restorations corresponding to the latter example (see Figures 4.6(h) and 4.6(l), and 4.6(g) and 4.6(k)) are more severely degraded by underestimating $M_f(\mathbf{x})$ and overestimating $M_h(\mathbf{x})$ than those of the former example (see Figures 4.5(c) and 4.5(f),

and 4.5(b) and 4.5(e)).

The investigation presented in this section confirms the importance of correct support estimation in achieving good results with the Davey Algorithm. In §4.3 different error measures are investigated in order to assist in the determination of the "best" support estimate.

4.2 THE FOURIER SPACE FILTER/CONSTRAINT

Having a similar form to the Wiener filter in (3.9), the Wiener-like filter in (4.3) is often expected to possess analogous features to the former. However, some observations of the behaviour of the algorithm suggest otherwise. Davey (1989, §7.4.3) reports that for a given value of the filter constant β, the tendency of the algorithm is to start with fast convergence. The subsequent iterations exhibit much slower convergence, and often the algorithm may eventually diverge. Such behaviour of the algorithm is also shown in Figure 4.4. This observation suggests that the value of β needs to be varied with iterations in order to achieve better performance of the algorithm, especially in situations where the value of β is selected too large. However, if the filter constant β is only related to the noise-to-signal ratio $|C(\mathbf{u})/F(\mathbf{u})|^2$ (the basis of the Wiener filter), which does not change as iterations proceed, what is the justification to vary β during the iterations? To verify whether the Wiener-like filter belongs to the same family as the Wiener, derivation of the least squares filter for blind deconvolution is presented in §4.2.1. The roles of the Wiener-like filter and its constant β are consequently discussed in §4.2.2.

Another point of discussion in the Wiener-like filter is the choice for n in (4.3). The effect of n is discussed in §4.2.3.

Seldin and Fienup (1990) comment that a different form of the filter should be used as the Davey's Wiener-like filter (see the comments immediately following Reference 6 of Seldin and Fienup (1990)). Results of performing the Davey algorithm presented in §4.1 indicate that the form of this filter published (Davey *et al.*, 1989) was correct, however. Seldin and Fienup's argument is discussed in §4.2.4, wherein the differences of the Davey's Wiener-like filter and the suggested filter are analyzed and comparative examples of using the two filters are presented.

It should be noted that further discussion and extensive studies of the Davey Algorithm in the rest of the section are only focused on the second half of the iterative loop in Figure 4.1, which generates new estimate $f_i(\mathbf{x})$ from $\hat{h}_i(\mathbf{x})$ and $g(\mathbf{x})$. A similar process is carried out at the first half of the loop where the roles of $f_i(\mathbf{x})$ and $\hat{h}_i(\mathbf{x})$ are replaced by that of $h_i(\mathbf{x})$ and $\hat{f}_i(\mathbf{x})$.

4.2.1 The least squares filter for blind deconvolution

The relationship of the least squares filters for standard deconvolution (i.e the standard Wiener filter in (3.6)) and for blind deconvolution is investigated in this section. To find out the form of this filter for blind deconvolution, the following derivation is deduced. This derivation is based on the treatment for standard deconvolution given by Rosenfeld and Kak (1982, §7.3) and is for real and two-dimensional images.

Considering blind deconvolution based on an iterative technique, since both image $f(\mathbf{x})$ and psf $h(\mathbf{x})$ are unknown *a priori*, the process usually starts with a very poor estimate of $f(\mathbf{x})$ or $h(\mathbf{x})$, for example, a pseudo-random image. The estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ are recovered from iteration-to-iteration until the "best" estimates are obtained. To derive a least squares filter in such a circumstance, the image estimate at each iteration, $f_i(\mathbf{x})$, is generated to minimize (in a statistical sense) the mean-square error with respect to the true image $f(\mathbf{x})$.

Since the minimization process involved is established in a statistical sense, the following derivation is approached in a statistical manner. Each pixel of a $N \times N$ image is considered to be an independent random variable. The N^2 independent random variables form a random field and are used to represent an image (Rosenfeld and Kak 1982, §2.4).

Let the blurred image, the true image and the contamination belong to the random fields $g(\mathbf{x})$, $f(\mathbf{x})$ and $c(\mathbf{x})$ respectively. The above mentioned mean-squared error can be expressed by

$$e_{f_i}^2 = \mathcal{E}[(f(\mathbf{x}) - f_i(\mathbf{x}))^2], \tag{4.7}$$

where $\mathcal{E}$ represents expected (average) value. Equation (4.7) is subject to the constraint that the estimate $f_i(\mathbf{x})$ be a linear function of the gray levels in the given blurred image $g(\mathbf{x})$,

$$f_i(\mathbf{x}) = \int l_i(\mathbf{x}, \mathbf{x}')g(\mathbf{x}')d\mathbf{x}', \tag{4.8}$$

where $l_i(\mathbf{x}, \mathbf{x}')$ is a linear function of $g(\mathbf{x})$ at the pixel $\mathbf{x}'$ and is to be determined under the condition that equation (4.7) is minimized. $g(\mathbf{x})$ can be described by

$$g(\mathbf{x}) = \int h(\mathbf{x} - \mathbf{x}')f(\mathbf{x}')d\mathbf{x}' + c(\mathbf{x}) \tag{4.9}$$

according to the degradation model in (1.5) and the property of random fields (see Rosenfeld and Kak, 1982, §2.4.5). Alternatively, $g(\mathbf{x})$ can be depicted by the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ at i^{th} iteration according to the iterative loop of the Davey Algorithm in Figure 4.1.

$$g(\mathbf{x}) = \int \hat{h}_i(\mathbf{x} - \mathbf{x}')f_i(\mathbf{x}')d\mathbf{x}' + \hat{c}_i(\mathbf{x}), \tag{4.10}$$

or

$$g(\mathbf{x}) = \int \hat{h}_i(\mathbf{x} - \mathbf{x}')\hat{f}_i(\mathbf{x}')d\mathbf{x}' + c_i(\mathbf{x}) \tag{4.11}$$

where $\hat{f}_i(\mathbf{x})$ is the input estimate of $f(\mathbf{x})$ at i^{th} iteration, $f_i(\mathbf{x})$ and $\hat{h}_i(\mathbf{x})$ are the new estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ separately at the iteration, $\hat{c}_i(\mathbf{x})$ and $c_i(\mathbf{x})$ are the respective residuals.

If the random fields are homogeneous, $l_i(\mathbf{x}, \mathbf{x}')$ should only depend on $(\mathbf{x} - \mathbf{x}')$ (Rosenfeld and Kak, 1982, §7.3), therefore equation (4.8) can be written as

$$f_i(\mathbf{x}) = \int l_i(\mathbf{x} - \mathbf{x}')g(\mathbf{x}')d\mathbf{x}'. \tag{4.12}$$

It can be shown that if a function $l_i(\mathbf{x} - \mathbf{x}')$ satisfies

$$\mathcal{E}[(f(\mathbf{x}) - \int l_i(\mathbf{x} - \mathbf{x}')g(\mathbf{x}')d\mathbf{x}')g(\tilde{\mathbf{x}})] = 0 \tag{4.13}$$

for all pixels $\mathbf{x}'$ and $\tilde{\mathbf{x}}$ in a two-dimensional space, $e^2_{\hat{f}_i}$ will be minimized (see Rosenfeld and Kak, 1982, §7.3).

It is convenient to define the autocorrelation and cross-correlation of the random field here. The autocorrelation of a random field $p(\mathbf{x})$ is defined by

$$r_{pp}(\mathbf{x}, \mathbf{x}') = \mathcal{E}[p(\mathbf{x})p(\mathbf{x}')]. \tag{4.14}$$

The cross-correlation of two random fields $p(\mathbf{x})$ and $q(\mathbf{x})$ is defined by

$$r_{pq}(\mathbf{x}, \mathbf{x}') = \mathcal{E}[p(\mathbf{x})q(\mathbf{x}')]. \tag{4.15}$$

If the random fields involved are homogeneous, that is the expected value of the random field is independent of position $\mathbf{x}$, equations (4.14) and (4.15) become

$$r_{pp}(\mathbf{x}, \mathbf{x}') = r_{pp}(\mathbf{x} - \mathbf{x}') \tag{4.16}$$

and

$$r_{pq}(\mathbf{x}, \mathbf{x}') = r_{pq}(\mathbf{x} - \mathbf{x}'). \tag{4.17}$$

(see Rosenfeld and Kak, 1982, §2.4.3).

Equation (4.13) can be rewritten as

$$\int l_i(\mathbf{x} - \mathbf{x}')\mathcal{E}[g(\mathbf{x}')g(\tilde{\mathbf{x}})]d\mathbf{x}' = \mathcal{E}[f(\mathbf{x})g(\tilde{\mathbf{x}}), \tag{4.18}$$

and substituting from equations (4.16) and (4.17),

$$\int l_i(\mathbf{x} - \mathbf{x}')r_{gg}(\mathbf{x}' - \tilde{\mathbf{x}})d\mathbf{x}' = r_{fg}(\mathbf{x} - \tilde{\mathbf{x}}). \tag{4.19}$$

Letting $\mathbf{x}' - \tilde{\mathbf{x}} = \mathbf{t}$ and $\mathbf{x} - \tilde{\mathbf{x}} = \tau$ gives

$$\int l_i(\tau - \mathbf{t})r_{gg}(\mathbf{t})d\mathbf{t} = r_{fg}(\tau). \tag{4.20}$$

Taking the Fourier transform of both sides of (4.20) and invoking the convolution theorem, it follows that

$$L_i(\mathbf{u})R_{gg}(\mathbf{u}) = R_{fg}(\mathbf{u}), \tag{4.21}$$

where $L_i(\mathbf{u})$ is the least squares filter at i^{th} iteration. $R_{gg}(\mathbf{u})$ is the autospectrum of the blurred image $g(\mathbf{x})$, and $R_{fg}(\mathbf{u})$ is the cross-spectrum of the true image and the blurred images.

To derive $R_{fg}(\mathbf{u})$, the cross-correlation of $f(\mathbf{x})$ and $g(\mathbf{x})$ is firstly formed from equations (4.11) and (4.15),

$$\begin{aligned} r_{fg}(\mathbf{x},\tilde{\mathbf{x}}) &= \mathcal{E}[f(\mathbf{x})g(\tilde{\mathbf{x}})] \\ &= \int \hat{h}_i(\tilde{\mathbf{x}}-\mathbf{x}')\mathcal{E}[f(\mathbf{x})\hat{f}_i(\mathbf{x}')]d\mathbf{x}' + \mathcal{E}[f(\mathbf{x})c_i(\tilde{\mathbf{x}})]. \end{aligned} \tag{4.22}$$

Making use of the homogeneity of the random fields in (4.17), equation (4.22) can be described as

$$r_{fg}(\mathbf{x}-\tilde{\mathbf{x}}) = \int \hat{h}_i(\tilde{\mathbf{x}}-\mathbf{x}')r_{f\hat{f}_i}(\mathbf{x}-\mathbf{x}')d\mathbf{x}' + r_{fc_i}(\mathbf{x}-\tilde{\mathbf{x}}). \tag{4.23}$$

By using substitutions similar to those employed in deriving (4.20) from (4.19), (4.23) can be written as

$$r_{fg}(\mathbf{t}) = \int \hat{h}_i(\tau-\mathbf{t})r_{f\hat{f}_i}(\tau)d\tau + r_{fc_i}(\mathbf{t}) \tag{4.24}$$

or

$$r_{fg}(\mathbf{t}) = \hat{h}_i(-\mathbf{t}) \odot r_{f\hat{f}_i}(\mathbf{t}) + r_{fc_i}(\mathbf{t}). \tag{4.25}$$

The cross-spectrum of $f(\mathbf{x})$ and $g(\mathbf{x})$ are obtained by equating Fourier transforms of both sides of (4.25), i.e.

$$R_{fg}(\mathbf{u}) = \hat{H}_i^*(\mathbf{u})R_{f\hat{f}_i}(\mathbf{u}) + R_{fc_i}(\mathbf{u}). \tag{4.26}$$

To derive $R_{gg}(\mathbf{u})$, the other quantity appearing in (4.21), equation (4.11) can be rewritten

$$g(\mathbf{x}) = \int \hat{f}_i(\mathbf{x}-\mathbf{x}')\hat{h}_i(\mathbf{x}')d\mathbf{x}' + c_i(\mathbf{x}). \tag{4.27}$$

Replacing $\mathbf{x}$ by $\mathbf{x}+\mathbf{a}$, (4.27) becomes

$$g(\mathbf{x}+\mathbf{a}) = \int \hat{f}_i(\mathbf{x}+\mathbf{a}-\mathbf{x}')\hat{h}_i(\mathbf{x}')d\mathbf{x}' + c_i(\mathbf{x}+\mathbf{a}). \tag{4.28}$$

The autocorrelation of $g(\mathbf{x})$ can be obtained by multiplying both sides of (4.28) by $g(\mathbf{a})$, taking the expectation and interchanging the order of integration and expectation, that is

$$r_{gg}(\mathbf{x}) = \int r_{\hat{f}_i g}(\mathbf{x}-\mathbf{x}')\hat{h}_i(\mathbf{x}')d\mathbf{x}' + r_{c_i g}(\mathbf{x}). \tag{4.29}$$

Symbolically, equation (4.29) can be expressed as

$$r_{gg}(\mathbf{x}) = r_{\hat{f}_i g}(\mathbf{x}) \odot \hat{h}_i(\mathbf{x}) + r_{c_i g}(\mathbf{x}), \tag{4.30}$$

where $r_{c_i g}(\mathbf{x})$ can be formed by replacing $\mathbf{x}$ in (4.27) by $\mathbf{a}$, multiplying both sides of (4.27) by $c_i(\mathbf{x}+\mathbf{a})$ and taking the expectation. That is

$$r_{c_i g}(\mathbf{x}) = \int r_{c_i \hat{f}_i}(\mathbf{x}+\mathbf{x}')\hat{h}_i(\mathbf{x}')d\mathbf{x}' + r_{c_i c_i}(\mathbf{x}), \tag{4.31}$$

which can be written as

$$r_{c_i g}(\mathbf{x}) = r_{c_i \hat{f}_i}(\mathbf{x}) \odot \hat{h}_i(-\mathbf{x}) + r_{c_i c_i}(\mathbf{x}). \tag{4.32}$$

Replacing $\mathbf{x}$ in (4.27) by $\mathbf{a}$, multiplying both sides of this equation by $\hat{f}_i(\mathbf{x}+\mathbf{a})$ and taking the expectation, $r_{\hat{f}_i g}(\mathbf{x})$ is obtained as

$$r_{\hat{f}_i g}(\mathbf{x}) = \int r_{\hat{f}_i \hat{f}_i}(\mathbf{x}+\mathbf{x}')\hat{h}_i(\mathbf{x}')d\mathbf{x}' + r_{\hat{f}_i c_i}(\mathbf{x}), \tag{4.33}$$

that is

$$r_{\hat{f}_i g}(\mathbf{x}) = r_{\hat{f}_i \hat{f}_i}(\mathbf{x}) \odot \hat{h}_i(-\mathbf{x}) + r_{\hat{f}_i c_i}(\mathbf{x}). \tag{4.34}$$

The autospectrum of $g(\mathbf{x})$ can be calculated by substituting (4.32) and (4.34) in (4.30) and using the Fourier transform, i.e.

$$R_{gg}(\mathbf{u}) = R_{\hat{f}_i \hat{f}_i}(\mathbf{u})|\hat{H}_i(\mathbf{u})|^2 + R_{\hat{f}_i c_i}(\mathbf{u})\hat{H}_i(\mathbf{u}) + R_{c_i \hat{f}_i}(\mathbf{u})\hat{H}_i^*(\mathbf{u}) + R_{c_i c_i}(\mathbf{u}). \tag{4.35}$$

Finally, substituting (4.26) and (4.35) in (4.21), the least squares filter at i^{th} iteration is obtained as

$$L_i(\mathbf{u}) = \frac{\hat{H}_i^*(\mathbf{u})R_{f\hat{f}_i}(\mathbf{u}) + R_{fc_i}(\mathbf{u})}{R_{\hat{f}_i \hat{f}_i}(\mathbf{u})|\hat{H}_i(\mathbf{u})|^2 + R_{\hat{f}_i c_i}(\mathbf{u})\hat{H}_i(\mathbf{u}) + R_{c_i \hat{f}_i}(\mathbf{u})\hat{H}_i^*(\mathbf{u}) + R_{c_i c_i}(\mathbf{u})}, \tag{4.36}$$

where $R_{c_i c_i}(\mathbf{u})$ and $R_{\hat{f}_i \hat{f}_i}(\mathbf{u})$ are the autospectra of the residual $c_i(\mathbf{x})$ and the image estimate $\hat{f}_i(\mathbf{x})$ at i^{th} iteration respectively. $R_{\hat{f}_i c_i}(\mathbf{u})$ and $R_{c_i \hat{f}_i}(\mathbf{u})$ are the cross-spectra of $\hat{f}_i(\mathbf{x})$ and $c_i(\mathbf{x})$ and vice versa.

The least squares filter for blind deconvolution in (4.36) appears to be much more complicated than its form in standard deconvolution in (3.6). The main reason is that the Wiener filter in standard deconvolution is simplified by the assumption that the true image $f(\mathbf{x})$ and the contamination $c(\mathbf{x})$ are uncorrelated and either $f(\mathbf{x})$ or $c(\mathbf{x})$ has zero mean (Rosenfeld and Kak, 1982, p284), that is

$$r_{fc}(\mathbf{x}) = \mathcal{E}[f(\mathbf{x})c(\mathbf{x})] = \mathcal{E}[f(\mathbf{x})]\mathcal{E}[c(\mathbf{x})] = 0, \tag{4.37}$$

and so

$$R_{fc}(\mathbf{u}) = 0. \tag{4.38}$$

However in blind deconvolution process, the cross-spectra $R_{fc_i}(\mathbf{u})$, $R_{\hat{f}_i c_i}(\mathbf{u})$ and $R_{c_i \hat{f}_i}(\mathbf{u})$ in (4.36), which are the Fourier transforms of $r_{fc_i}(\mathbf{x})$, $r_{\hat{f}_i c_i}(\mathbf{x})$ and $r_{c_i \hat{f}_i}(\mathbf{x})$, respectively, cannot be ignored even when $r_{fc}(\mathbf{x})$ is equal to zero. This may be proven by the following description.

$$\begin{aligned}
r_{fc_i}(\mathbf{x}) &= \mathcal{E}[f(\mathbf{x}+\mathbf{a})c_i(\mathbf{a})] \\
&= \mathcal{E}[f(\mathbf{x}+\mathbf{a})[g(\mathbf{a}) - \hat{f}_i(\mathbf{a}) \odot \hat{h}_i(\mathbf{a})]] \\
&= \mathcal{E}[f(\mathbf{x}+\mathbf{a})[\int f(\mathbf{a}-\mathbf{x}')h(\mathbf{x}')d\mathbf{x}' + c(\mathbf{a}) - \int \hat{f}_i(\mathbf{a}-\mathbf{x}')\hat{h}_i(\mathbf{x}')d\mathbf{x}']] \\
&= \int r_{ff}(\mathbf{x}+\mathbf{x}')h(\mathbf{x}')d\mathbf{x}' + r_{fc}(\mathbf{x}) - \int r_{f\hat{f}_i}(\mathbf{x}+\mathbf{x}')\hat{h}_i(\mathbf{x}')d\mathbf{x}' \\
&= r_{ff}(\mathbf{x}) \odot h(-\mathbf{x}) + r_{fc}(\mathbf{x}) - r_{f\hat{f}_i}(\mathbf{x}) \odot \hat{h}_i(-\mathbf{x}).
\end{aligned} \tag{4.39}$$

Taking the Fourier transform of both sides of equation (4.39), then

$$R_{fc_i}(\mathbf{u}) = R_{ff}(\mathbf{u})H^*(\mathbf{u}) + R_{fc}(\mathbf{u}) - R_{f\hat{f}_i}(\mathbf{u})\hat{H}_i^*(\mathbf{u}). \tag{4.40}$$

Similarly,

$$r_{\hat{f}_i c_i}(\mathbf{x}) = r_{\hat{f}_i f}(\mathbf{x}) \odot h(-\mathbf{x}) + r_{\hat{f}_i c}(\mathbf{x}) - r_{\hat{f}_i \hat{f}_i}(\mathbf{x}) \odot \hat{h}_i(-\mathbf{x}), \tag{4.41}$$

and

$$R_{\hat{f}_i c_i}(\mathbf{u}) = R_{\hat{f}_i f}(\mathbf{u})H^*(\mathbf{u}) + R_{\hat{f}_i c}(\mathbf{u}) - R_{\hat{f}_i \hat{f}_i}(\mathbf{u})\hat{H}_i^*(\mathbf{u}). \tag{4.42}$$

Since the cross-spectrum of two random fields $p(\mathbf{x})$ and $q(\mathbf{x})$ satisfies (Bendat and Piersol, 1980, §3.2.1)

$$R_{pq}(\mathbf{u}) = R_{qp}(-\mathbf{u}) \tag{4.43}$$

and

$$R_{pq}(\mathbf{u}) = P(-\mathbf{u})Q(\mathbf{u}), \tag{4.44}$$

$R_{c_i \hat{f}_i}(\mathbf{u})$ can be expressed by

$$\begin{aligned} R_{c_i \hat{f}_i}(\mathbf{u}) &= R_{\hat{f}_i c_i}(-\mathbf{u}) \\ &= R_{f\hat{f}_i}(\mathbf{u})H(\mathbf{u}) + R_{c\hat{f}_i}(\mathbf{u}) - R_{\hat{f}_i \hat{f}_i}(\mathbf{u})\hat{H}_i(\mathbf{u}). \end{aligned} \tag{4.45}$$

Equations (4.40), (4.42) and (4.45) show that the sizes of each of $R_{fc_i}(\mathbf{u})$, $R_{\hat{f}_i c_i}(\mathbf{u})$ and $R_{c_i \hat{f}_i}(\mathbf{u})$ are dependent on quantities which relate to the difference between the spectral estimates $\hat{F}_i(\mathbf{u})$ and $\hat{H}_i(\mathbf{u})$ and their true versions $F(\mathbf{u})$ and $H(\mathbf{u})$. For example in (4.42) the difference between the first and third terms becomes less as $\hat{F}_i(\mathbf{u})$ and $\hat{H}_i(\mathbf{u})$ converge towards their solutions. These terms cannot be neglected, however, especially near the beginning of the iterative process, because the initial estimates of the image and psf are poor and similarly their spectra are very different from the actual spectra of the true image and psf respectively.

The sizes of $R_{fc_i}(\mathbf{u})$, $R_{\hat{f}_i c_i}(\mathbf{u})$ and $R_{c_i \hat{f}_i}(\mathbf{u})$ are also related to the contamination involved in the blurred image due the presence of $R_{fc}(\mathbf{u})$, $R_{\hat{f}_i c}(\mathbf{u})$ and $R_{c\hat{f}_i}(\mathbf{u})$ in (4.40), (4.42) and (4.45) respectively. If $f(\mathbf{x})$ and $c(\mathbf{x})$ are uncorrelated and either of them has zero mean, $R_{fc}(\mathbf{u})$ can be neglected as shown in (4.37) and (4.38). However, this does not guarantee $R_{\hat{f}_i c}(\mathbf{u})$ and $R_{c\hat{f}_i}(\mathbf{u})$ to be negligible in size. In fact $\hat{f}_i(\mathbf{x})$ is dependent on $c(\mathbf{x})$ which can be shown by equating the RHS of (4.9) and (4.11). Their dependence may be easier to see by their spectra. Taking the Fourier transform of the right sides of (4.9) and (4.11) and equating them, $\hat{F}_i(\mathbf{u})$ can be expressed by

$$\hat{F}_i(\mathbf{u}) = \frac{F(\mathbf{u})H(\mathbf{u})}{\hat{H}_i(\mathbf{u})} + \frac{C(\mathbf{u})}{\hat{H}_i(\mathbf{u})} - \frac{C_i(\mathbf{u})}{\hat{H}_i(\mathbf{u})}, \tag{4.46}$$

where it is assumed that only regions of Fourier space for which $|\hat{H}_i(\mathbf{u})| \neq 0$ are being considered. According to (4.44) and (4.46), the cross-spectrum of $\hat{f}_i(\mathbf{x})$ and $c(\mathbf{x})$ can be formed by

$$\begin{aligned} R_{\hat{f}_i c}(\mathbf{u}) &= \hat{F}_i(-\mathbf{u})C(\mathbf{u}) \\ &= \frac{F(-\mathbf{u})H(-\mathbf{u})C(\mathbf{u})}{\hat{H}_i(-\mathbf{u})} + \frac{C(-\mathbf{u})C(\mathbf{u})}{\hat{H}_i(-\mathbf{u})} - \frac{C_i(-\mathbf{u})C(\mathbf{u})}{\hat{H}_i(-\mathbf{u})}, \end{aligned} \tag{4.47}$$

where $F(-\mathbf{u})C(\mathbf{u})$ in the first term is the cross-spectrum of $f(\mathbf{x})$ and $c(\mathbf{x})$ and is equal to zero according to (4.38). Replacing the numerators of the second and third terms of (4.47) by $R_{cc}(\mathbf{u})$ and $R_{c_i c}(\mathbf{u})$ separately, (4.47) becomes

$$R_{\hat{f}_i c}(\mathbf{u}) = \frac{R_{cc}(\mathbf{u})}{\hat{H}_i(\mathbf{u})} - \frac{R_{c_i c}(\mathbf{u})}{\hat{H}_i(\mathbf{u})}. \tag{4.48}$$

Equation (4.48) reveals that $R_{\hat{f}_i c}(\mathbf{u})$ cannot be neglected since $C_i(\mathbf{u})$ is different from $C(\mathbf{u})$ in general. Therefore $\hat{f}_i(\mathbf{x})$ and $c(\mathbf{x})$ are correlated.

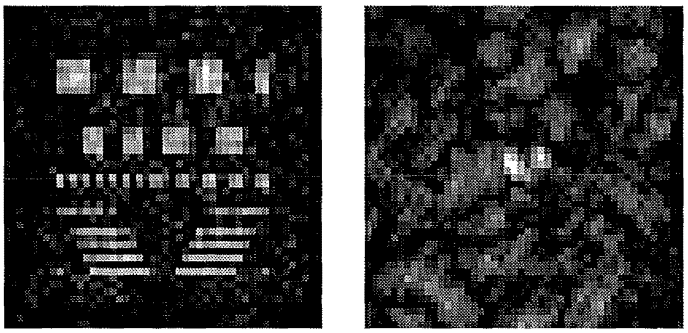
The above discussion from (4.37) to (4.48) shows that the terms of cross-spectra $R_{fc_i}(\mathbf{u})$, $R_{\hat{f}_i c_i}(\mathbf{u})$ and $R_{c_i \hat{f}_i}(\mathbf{u})$ in the least squares filter (4.36) result from the contamination in $g(\mathbf{x})$ and the differences between the spectra, $F(\mathbf{u})$ and $\hat{F}_i(\mathbf{u})$, $H(\mathbf{u})$ and $\hat{H}_i(\mathbf{u})$. They are not equal to zero, so their related terms in (4.36) cannot be ignored in general. Therefore the least squares filter in blind deconvolution given in (4.36) cannot be further simplified. It differs significantly from both the standard Wiener filter in (3.9) and the Wiener-like filter in (4.3).

Note that the least squares filter in (4.36) is related to the Davey's Wiener-like filter in (4.3) for $n = 0$ in the following case. As the iterative process proceeds, the algorithm may converge to the solution step by step. The difference between a spectrum and its estimate, say $F(\mathbf{u})$ and $\hat{F}_i(\mathbf{u})$ or $H(\mathbf{u})$ and $\hat{H}_i(\mathbf{u})$, is getting smaller. The sizes of $R_{fc_i}(\mathbf{u})$, $R_{\hat{f}_i c_i}(\mathbf{u})$ and $R_{c_i \hat{f}_i}(\mathbf{u})$ in (4.40), (4.42) and (4.45) are decreasing thereby. $R_{fc_i}(\mathbf{u})$, $R_{\hat{f}_i c_i}(\mathbf{u})$ and $R_{c_i \hat{f}_i}(\mathbf{u})$ will vanish when both spectral estimates $\hat{F}_i(\mathbf{u})$ and $\hat{H}_i(\mathbf{u})$ have converged close to their solutions $F(\mathbf{u})$ and $H(\mathbf{u})$. In other words, when both $\hat{f}_i(\mathbf{x})$ and $\hat{h}_i(\mathbf{x})$ are faithful restorations of their true versions, and $f(\mathbf{x})$ and $c(\mathbf{x})$ satisfy (4.37), then $R_{fc_i}(\mathbf{u})$, $R_{\hat{f}_i c_i}(\mathbf{u})$ and $R_{c_i \hat{f}_i}(\mathbf{u})$ in (4.36) become small and negligible. Equation (4.36) will reduce to a form of

$$W_{\hat{H}_i}(\mathbf{u}) = \frac{\hat{H}_i^*(\mathbf{u})}{|\hat{H}_i(\mathbf{u})|^2 + \Phi(\mathbf{u})}, \tag{4.49}$$

where $\Phi(\mathbf{u}) = R_{c_i c_i}(\mathbf{u})/R_{\hat{f}_i \hat{f}_i}(\mathbf{u}) = |C_i(\mathbf{u})|^2/|\hat{F}_i(\mathbf{u})|^2$ or $\Phi(\mathbf{u}) = |C(\mathbf{u})|^2/|F(\mathbf{u})|^2$ according to the condition which equation (4.49) is based on. If $\Phi(\mathbf{u})$ is approximated by a constant β, equation (4.49) becomes the same as the Wiener-like filter in (4.3) for $n = 0$. Unfortunately, the process of the iterative blind deconvolution generally does not satisfy the condition to obtain (4.49) from (4.36). The concept of blind deconvolution is defined under the condition that neither $h(\mathbf{x})$ nor $f(\mathbf{x})$ is known *a priori*, and the

process of the iterative blind deconvolution usually starts with a random image, which is very much different from its true version.



(a) (b)

Figure 4.8 Demonstration of blind deconvolution using the least squares filter in (4.36), (a) and (b) restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$.

The effect of the least squares filter in (4.36) on blind deconvolution is now demonstrated. An iterative algorithm is formed by replacing Davey's Wiener-like filter in (4.3) by the least squares filter in (4.36) wherein $R_{f\hat{f}_i}$ is approximated by $R_{\hat{f}_i\hat{f}_i}$. A similar substitution of the filters is applied when the estimate of $H(\mathbf{u})$ is to be generated. The rest of this iterative algorithm is the same as the Davey Algorithm described in Figure 4.1. This iterative algorithm was applied to the blurred image $g(\mathbf{x})$ shown in Figure 4.2(c), where $g(\mathbf{x})$ was contaminated with SNR of 30dB. The pseudo-random image shown in Figure 4.2(d) was used and the initial support estimates of $M_f(\mathbf{x}) = B_f(\mathbf{x})$ and $M_h(\mathbf{x}) = B_h(\mathbf{x})$ were selected. Restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ are shown in Figures 4.8(a) and 4.8(b) which were obtained at the iteration where $E_T[f_i]$ was minimum within maximum 100 iterations.

Inspection of Figure 4.8 shows that successful blind deconvolution was achieved using the least squares filter derived in (4.36). $f(\mathbf{x})$ can be seen in Figure 4.8(a) to have been well restored. An equivalent quality of recovery for $h(\mathbf{x})$ did not occur, however, until some iterations latter.

To summarize the section, the derivation of the least squares filter in the blind deconvolution reveals that since the condition for blind deconvolution is different from that for standard deconvolution, the forms of the least squares filter for the two cases appear to be quite different. The least squares filter for blind deconvolution involves the cross-spectrum of $f(\mathbf{x})$ and $c_i(\mathbf{x})$, as well as that of $\hat{f}_i(\mathbf{x})$ and $c_i(\mathbf{x})$, which have been shown to be correlated even when $f(\mathbf{x})$ and $c(\mathbf{x})$ are uncorrelated; the related terms cannot therefore be neglected. When both estimates $f_i(\mathbf{x})$ and $\hat{h}_i(\mathbf{x})$ are recovered to be faithful restorations of their true versions, $R_{fc_i}(\mathbf{u})$, $R_{\hat{f}_ic_i}(\mathbf{u})$ and $R_{c_i\hat{f}_i}(\mathbf{u})$ vanish to zero. Only then does the least squares filter in (4.36) reduce to the standard Wiener

filter (*cf.* (3.6)) and the Wiener-like filter in (4.3) for $n = 0$. The Wiener-like filter in (4.3) is not a least squares filter in general.

It should be noted that the derivation leading to (4.36) is carried out with respect to the image estimate $\hat{f}_i(\mathbf{x})$. A similar derivation arrives at the least squares filter with respect to the psf estimate $h_i(\mathbf{x})$. In that case the least squares filter is derived under the condition that it is the psf estimate at each iteration to be generated to minimize the mean-squared error of the true psf and the estimated psf.

4.2.2 The role of Wiener-like filter

Comparing the forms of the least squares filters for blind deconvolution in (4.36) and for standard deconvolution in (3.6) reveals two points: Firstly, the former filter, which contains the same optimal property as the standard Wiener filter, appears to have a very different form to the latter. Secondly, the Wiener-like filter in (4.3) for $n = 0$, which maintains the similar form to the standard Wiener filter, is not optimal from the least squares sense. In other words, the Wiener-like filter for blind deconvolution does not function the same as the Wiener filter for standard deconvolution. However, experiments show that satisfactory restorations can still be obtained by incorporating the Wiener-like filter in (4.3) into the iterative blind deconvolution algorithm. How then does this Wiener-like filter function in blind deconvolution, and what is the role of its filter constant β? These points are discussed in the rest of the section.

It is convenient to rewrite (4.3) in the form

$$W_{\hat{H}_i}(\mathbf{u}) = \frac{1}{[|\hat{H}_i(\mathbf{u})| + \beta/|\hat{H}_i(\mathbf{u})|^{n+1}]e^{i\mathcal{P}[\hat{H}_i]}}, \tag{4.50}$$

where $\mathcal{P}[\hat{H}_i]$ is the phase of $\hat{H}_i(\mathbf{u})$. Equation (4.50) appears as a modified form of an inverse filter.

The relationship of the Wiener-like filter in (4.50) to β and $\hat{H}_i(\mathbf{u})$ is depicted by plots of $|W_{\hat{H}_i}(\mathbf{u})|$ against $|H_i(\mathbf{u})|$ for a range of values of β in Figure 4.9 (n is here set to zero). Inspection of Figure 4.9 reveals: For a fixed value of β, $|W_{\hat{H}_i}(\mathbf{u})|$ asymptotically approaches the inverse filter when $|H_i(\mathbf{u})|$ is greater than but not close to a value H_{max}, which results in a maximum for $|W_{\hat{H}_i}(\mathbf{u})|$ marked by "x"; when $|\hat{H}_i(\mathbf{u})|$ is less than H_{max}, the magnitude of $W_{\hat{H}_i}(\mathbf{u})$ is increasingly suppressed as $|\hat{H}_i(\mathbf{u})|$ decreases, however, the phase of $W_{\hat{H}_i}(\mathbf{u})$ retains the same as that of the inverse filter (see (4.50)). For the same range values of $|\hat{H}_i(\mathbf{u})|$, H_{max} varies directly as β.

The above observation shows that H_{max} is a critical value for $|\hat{H}_i(\mathbf{u})|$ in $W_{\hat{H}_i}(\mathbf{u})$. The relationship of $|W_{\hat{H}_i}(\mathbf{u})|$ and $|\hat{H}_i(\mathbf{u})|$ changes at H_{max} from obeying the inverse filter to attenuating the magnitude of the inverse filter. In other words, H_{max} acts as a cut-off spectral magnitude which separates $|\hat{H}_i(\mathbf{u})|$ into two portions. One (for which $|\hat{H}_i(\mathbf{u})| > H_{max}$) is actively involved in a process of restoring the new estimate by an inverse filter.

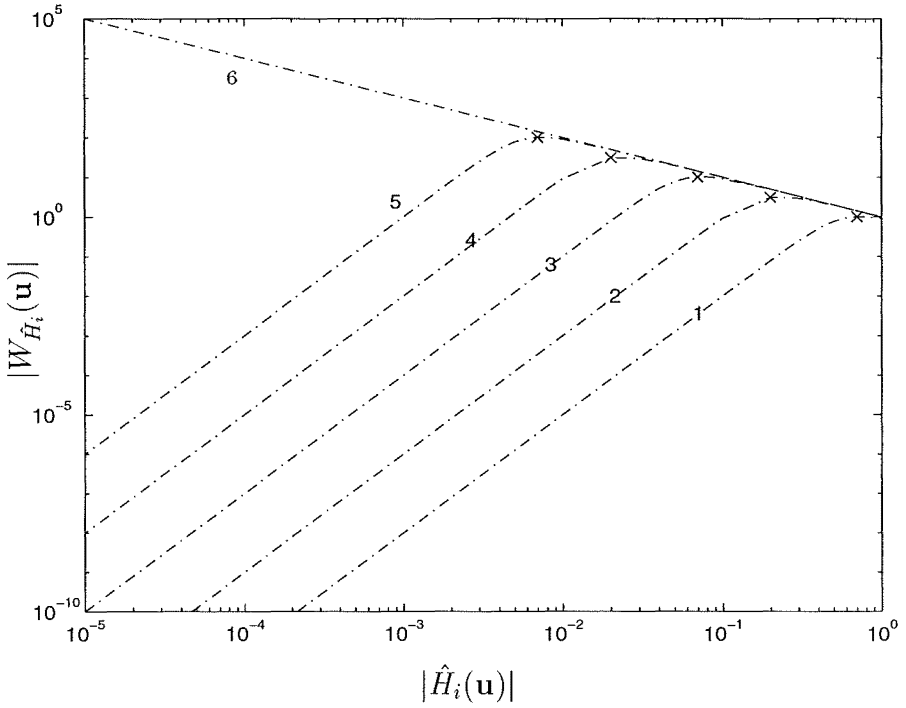


Figure 4.9 Plots of the Wiener-like filter $|W_{\hat{H}_i}(\mathbf{u})|$ versus $|\hat{H}_i(\mathbf{u})|$ for a range of β, both $|W_{\hat{H}_i}(\mathbf{u})|$ and $|\hat{H}_i(\mathbf{u})|$ are logarithmic scaled, curve 1: $\beta = 10^{-1}$; curve 2: $\beta = 10^{-3}$; curve 3: $\beta = 10^{-5}$; curve 4: $\beta = 10^{-7}$; curve 5: $\beta = 10^{-9}$; curve 6: $\beta = 0$ (i.e. an inverse filter). The horizontal coordinates of the symbols "x" stand for H_{max} for each curve.

The other (for which $|\hat{H}_i(\mathbf{u})| < H_{max}$) is essentially discarded from the point of view of restoration. These two portions of $\hat{H}_i(\mathbf{u})$ are denoted "effective" and "ineffective" respectively.

The value of H_{max} can be obtained by taking partial derivative of $|W_{\hat{H}_i}(\mathbf{u})|$ with respect to $|\hat{H}_i(\mathbf{u})|$,

$$\frac{\partial |W_{\hat{H}_i}(\mathbf{u})|}{\partial |\hat{H}_i(\mathbf{u})|} = -\frac{1-(1+n)\beta/|\hat{H}_i(\mathbf{u})|^{n+2}}{(|\hat{H}_i(\mathbf{u})| + \beta/|\hat{H}_i(\mathbf{u})|^{n+1})^2} \tag{4.51}$$

and equating the RHS of (4.51) to zero, that is

$$H_{max} = [(n+1)\beta]^{\frac{1}{n+2}}. \tag{4.52}$$

Equation (4.52) shows that the choice of β controls the value of H_{max} in force.

The size of H_{max} has significant influence on the quality of the restorations. Considering the effective part of $\hat{H}_i(\mathbf{u})$, the Wiener-like filter (*cf.* (4.50)) corresponding to this part of $\hat{H}_i(\mathbf{u})$ performs as an inverse filter, $W_{\hat{H}_i}(\mathbf{u}) \approx 1/\hat{H}_i(\mathbf{u})$. Consequently, the corresponding components of the estimate of $F(\mathbf{u})$ in (4.4) are generated by

$$F_i(\mathbf{u}) = \frac{G(\mathbf{u})}{\hat{H}_i(\mathbf{u})}. \tag{4.53}$$

Substituting for $G(\mathbf{u})$ from (2.12)

$$G(\mathbf{u}) = F(\mathbf{u})H(\mathbf{u}) + C(\mathbf{u}), \tag{4.54}$$

equation (4.53) becomes

$$F_i(\mathbf{u}) = \frac{F(\mathbf{u})H(\mathbf{u})}{\hat{H}_i(\mathbf{u})} + \frac{C(\mathbf{u})}{\hat{H}_i(\mathbf{u})}. \tag{4.55}$$

If H_{max} is selected too small, low magnitudes of $\hat{H}_i(\mathbf{u})$ can still become the effective components of $\hat{H}_i(\mathbf{u})$ and get involved in (4.55) to produce $F_i(\mathbf{u})$. Then, if the contamination related term in (4.55) is appreciably greater than its signal related term, that is $|C(\mathbf{u})|/|\hat{H}_i(\mathbf{u})| \gg |F(\mathbf{u})\,H(\mathbf{u})\,|/|\hat{H}_i(\mathbf{u})|$, or equivalently

$$|H(\mathbf{u})| \ll |C(\mathbf{u})/F(\mathbf{u})|, \tag{4.56}$$

the contamination in (4.55) will be amplified by these low magnitudes of $\hat{H}_i(\mathbf{u})$, and the corresponding components of $F_i(\mathbf{u})$ will be contamination dominated.

To overcome contamination amplification in the estimate $F_i(\mathbf{u})$, it is sensible to exclude those components of $\hat{H}_i(\mathbf{u})$ which are likely to overexaggerate contamination in $G(\mathbf{u})$, from the attempted restoration process in (4.55). Since $\hat{H}_i(\mathbf{u})$ approximates $H(\mathbf{u})$ and contamination amplification occurs when $H(\mathbf{u})$ satisfies equation (4.56), this suggests that this exclusion might be achieved by selecting a functional cut-off spectral magnitude, denoted $H_{max}(\mathbf{u})$, which is chosen to be proportional to the quantity at the right hand side of (4.56),

$$H_{max}(\mathbf{u}) = a|C(\mathbf{u})/F(\mathbf{u})|, \tag{4.57}$$

where a is a real constant ranging over $0 < a \leq 1$. All $\hat{H}_i(\mathbf{u})$ satisfying $|\hat{H}_i(\mathbf{u})| < H_{max}(\mathbf{u}) = a|C(\mathbf{u})/F(\mathbf{u})|$ are thereby rendered ineffective and removed from the operation in (4.55). Obviously, $H_{max}(\mathbf{u})$ is the best or "ideal" choice for H_{max}.

Since $H_{max}(\mathbf{u})$ is selected through β, the "ideal" quantity for β is consequently derived from (4.52) and (4.57) as a function of $\mathbf{u}$, that is

$$\beta(\mathbf{u}) = \frac{a^{n+2}}{n+1}\left|\frac{C(\mathbf{u})}{F(\mathbf{u})}\right|^{n+2}. \tag{4.58}$$

When $n = 0$ and $a = 1$ this reduces to $\beta(\mathbf{u}) = |C(\mathbf{u})|^2/|F(\mathbf{u})|^2$.

It should be noted that the ineffective part of $\hat{H}_i(\mathbf{u})$ separated by the "ideal" quantity $H_{max}(\mathbf{u})$ in (4.57) corresponds to the contamination dominated components of $G(\mathbf{u})$ (for which $|C(\mathbf{u})| \gg |H(\mathbf{u})F(\mathbf{u})|$). Consequently, the effective part of $\hat{H}_i(\mathbf{u})$ matches the useful components of $G(\mathbf{u})$.

Equations (4.57) and (4.58) show that the size of $H_{max}(\mathbf{u})$ and $\beta(\mathbf{u})$ are directly proportional to $|C(\mathbf{u})|$ and $|C(\mathbf{u})|^{n+2}$ separately. The more severely an image is contaminated, the larger the quantities for $H_{max}(\mathbf{u})$ and $\beta(\mathbf{u})$ required, consequently the larger the values for H_{max} and β needed.

When the Wiener-like filter is involved in the whole iterative blind deconvolution process, it is used as the Fourier space constraint to generate new spectral estimates. In this case, even when H_{max} is set "ideally", as described in (4.57), so that components of $G(\mathbf{u})$ involved in (4.53) are useful, it is still not sufficient to retrieve an good estimate of $F(\mathbf{u})$ in the early iterations. The estimate $\hat{H}_i(\mathbf{u})$, or at least most of its effective components ($|\hat{H}_i(\mathbf{u})| > H_{max}$) are also required to be effective or good estimates of $H(\mathbf{u})$. Unfortunately this is not the case to start with in blind deconvolution, where the initial estimate is usually chosen as a pseudo-random image. In the early iterations, many effective components of $\hat{H}_i(\mathbf{u})$ are very noisy. The new estimate $F_i(\mathbf{u})$ generated from $\hat{H}_i(\mathbf{u})$ in (4.55) is therefore seriously contaminated. However, the deconvolution process still converges. This is because the image space constraints urge the inverse Fourier transform of the estimate $F_i(\mathbf{u})$ to converge towards its solution. The quality of the estimate is improved thereby. The similar process occurs to $\hat{h}_i(\mathbf{x})$ at the same iteration. As iteration proceeds, more and more effective components of $\hat{H}_i(\mathbf{u})$ and $F_i(\mathbf{u})$ are retrieved gradually until all, to the limit imposed by H_{max} and F_{max} (corresponding to $W_{\hat{F}_i}(\mathbf{u})$), are estimated. The algorithm stops converging then and good estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ are restored, provided parameters are chosen wisely.

Figure 4.10 demonstrates estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ recorded in different stages of blind deconvolution process. The Davey Algorithm, wherein β in $W_{\hat{H}_i}(\mathbf{u})$ (see (4.50)) was replaced by the "ideal" quantity in (4.58) with $n = 0$ and $a = 0.5$, was applied to the blurred image shown in Figure 4.2(c). The support estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$, $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$, were selected to be equal to the image boxes $B_f(\mathbf{x})$ and $B_h(\mathbf{x})$ respectively. A pseudo-random image, shown in Figure 4.2(d), was used as an initial estimate of $f(\mathbf{x})$. Estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ obtained at iterations 1, 5, 10 and 20 are shown in Figure 4.10, and true errors of the estimates over 100 iterations are depicted in Figure 4.11. Observation of Figures 4.10 and 4.11 shows that the qualities of the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ are improved as iteration proceeds, and fast convergence of the algorithm occurs within first 20 iterations; this convergence stops once it reaches certain stage, which is governed by many factors described in §4.1. Note that when generating the estimate of $H(\mathbf{u})$ in the other half of the iterative loop, $\beta(\mathbf{u})$ for $W_{\hat{F}_i}(\mathbf{u})$ was formed by substituting $F(\mathbf{u})$ in (4.58) with $H(\mathbf{u})$.

In §4.1.1 the effect of β on the performance of the algorithm is demonstrated (see Figure 4.3). It is appropriate to discuss this effect here. Since $\beta(\mathbf{u})$ is usually unknown *a priori* in practice, the value of β is determined experimentally.

When β is selected to be too large, the related H_{max} is set too high. The Wiener-like filter in (4.50) allocates some useful components of $\hat{H}_i(\mathbf{u})$ in the range of ineffective part and, consequently, weakens the corresponding useful components of $G(\mathbf{u})$ in the estimate of $F(\mathbf{u})$. Since low magnitude of the spectrum is typically associated with high spatial frequency and the high frequency information contains a lot detail of the image (Biemond *et al.*, 1990), the cost of discarding useful low values of $|\hat{H}_i(\mathbf{u})|$ from a

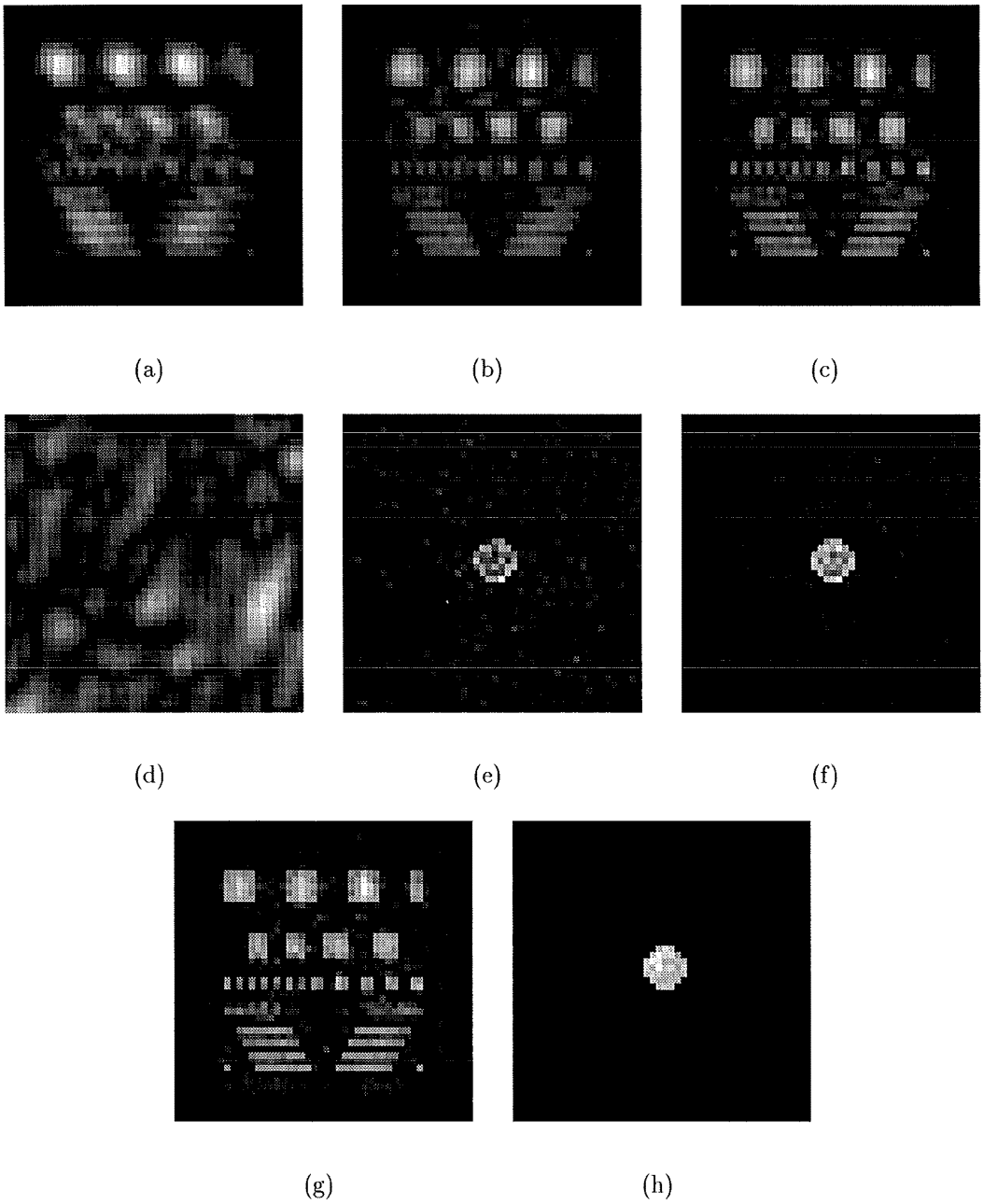


Figure 4.10 Demonstration of the blind deconvolution process using the Davey Algorithm with β replaced by $\beta(\mathbf{u})$, $a = 0.5$, $n = 0$, SNR = 30dB, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ obtained at iteration i, corresponding to Figures 4.2(a) and 4.2(b): (a) and (d) $i = 1$, (b) and (e) $i = 5$, (c) and (f) $i = 10$, (g) and (h) $i = 20$.

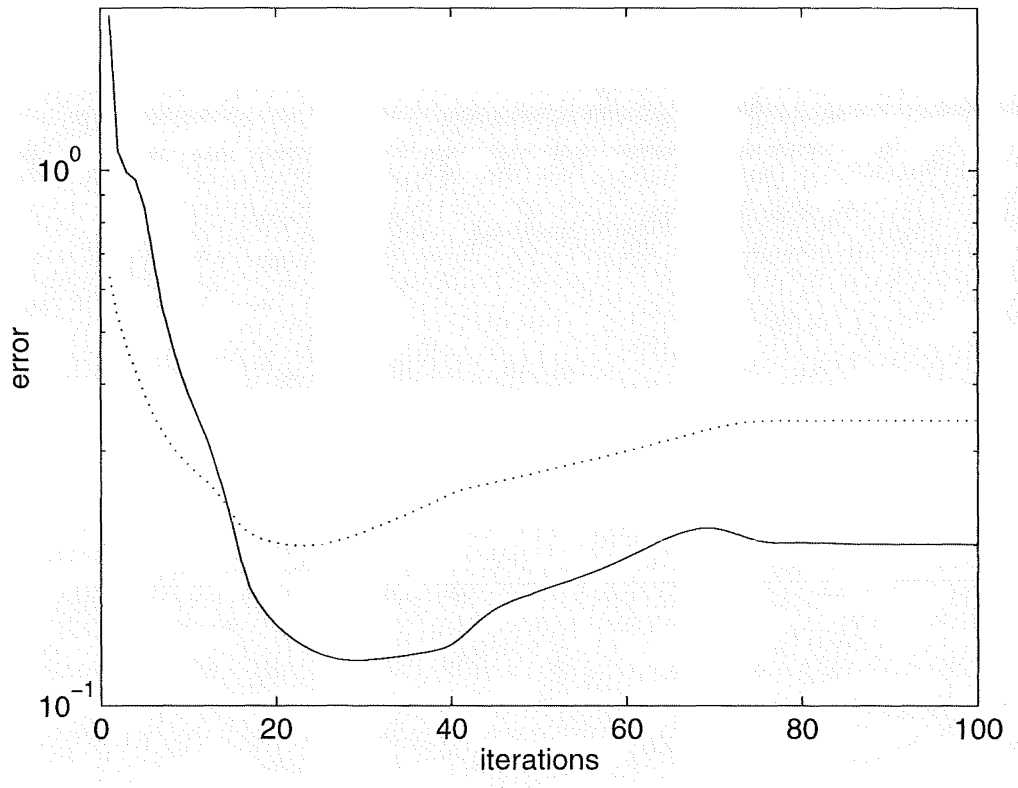


Figure 4.11 True errors of the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ corresponding to Figure 4.10, dotted and continuous curves represent $E_T[f_i]$ and $E_T[h_i]$ respectively.

deconvolution related process is to attenuate high frequency information of $F_i(\mathbf{u})$ and lose detail in the image estimate $f_i(\mathbf{x})$. The restoration is therefore a blurred version of the true image which is shown in Figure 4.3(g).

When β is chosen to be too small, H_{max} is set much lower than what required by its "ideal" quantity $H_{max}(\mathbf{u})$. Many components of $\hat{H}_i(\mathbf{u})$ with low magnitudes are assigned in the effective part of $\hat{H}_i(\mathbf{u})$, which causes the Wiener-like filter to involve many contamination dominated components of $G(\mathbf{u})$ with the low magnitudes of $\hat{H}_i(\mathbf{u})$ in a deconvolution process approximating to an inverse filtering. These contamination dominated components of $G(\mathbf{u})$ are thus overemphasized in the estimate of $F(\mathbf{u})$ which results in a spurious restoration of $f(\mathbf{x})$ as shown in Figure 4.3(e).

When β is suitably chosen, H_{max} is set to be compatible with $H_{max}(\mathbf{u})$ and divides the effective and ineffective components of $\hat{H}_i(\mathbf{u})$ in an appropriate proportion that matches well with usefulness of the components of $G(\mathbf{u})$. Such a separation of $\hat{H}_i(\mathbf{u})$ enables the Wiener-like filter to involve effective components of $G(\mathbf{u})$ with the inverse filter to produce useful information of $F(\mathbf{u})$, and reduces the effect of contamination dominated

components of $G(\mathbf{u})$ (for which $|F(\mathbf{u})H(\mathbf{u})| \ll |C(\mathbf{u})|$) on the estimate of $F(\mathbf{u})$. In other words, a suitable choice of β allows the Wiener-like filter to make good use of the information contained in $G(\mathbf{u})$ and, in the meantime, attenuates the influence of the contamination of $G(\mathbf{u})$ on the new estimate (*cf.* Figure 4.3(c)).

4.2.3 The role of the exponent n

The role of the exponent n in the performance of the Wiener-like filter (4.3) is discussed in this section. Since the Wiener-like filter (*cf.* (4.50)) is purely a modified inverse filter, it is sensible to investigate the performance of the filter for different values of n according to the following criteria: how closely the filter approximates to an inverse filter for the effective part of $\hat{H}_i(\mathbf{u})$ ($|\hat{H}_i(\mathbf{u})| \geq H_{max}$) and how efficiently the filter suppresses its magnitude for the ineffective part of $\hat{H}_i(\mathbf{u})$ ($|\hat{H}_i(\mathbf{u})| < H_{max}$). The discussion is therefore carried out in terms of two portions of $\hat{H}_i(\mathbf{u})$.

In the effective part of $\hat{H}_i(\mathbf{u})$ ($|\hat{H}_i(\mathbf{u})| \geq H_{max}$), the Wiener-like filter is reduced to an inverse filter when $|\hat{H}_i(\mathbf{u})| \gg \beta/|\hat{H}_i(\mathbf{u})|^{n+1}$, while its magnitude is deviated from that of the inverse filter $I_{\hat{H}_i}(\mathbf{u})$ as $|\hat{H}_i(\mathbf{u})|$ approaches H_{max}. The deviation of $|W_{\hat{H}_i}(\mathbf{u})|$ from $|I_{\hat{H}_i}(\mathbf{u})|$ is termed as bias (Davey, 1989, §7.4.1.2), which is due to the effect of the second term of the denominator in (4.50), expressed by

$$\beta/|\hat{H}_i(\mathbf{u})|^{1+n} = \frac{H_{max}}{1+n}\left(\frac{H_{max}}{|\hat{H}_i(\mathbf{u})|}\right)^{n+1}. \tag{4.59}$$

Since $|\hat{H}_i(\mathbf{u})| \geq H_{max}$ in the range of concern, then $H_{max}/|\hat{H}_i(\mathbf{u})| \leq 1$. The closer $|\hat{H}_i(\mathbf{u})|$ approaches H_{max}, the larger is the value of $\beta/|\hat{H}_i(\mathbf{u})|^{n+1}$, and the more severe is the bias, i.e. the more $|W_{\hat{H}_i}(\mathbf{u})|$ is away from $|I_{\hat{H}_i}(\mathbf{u})|$ as shown in Figure 4.12. Also evident in Figure 4.12 is that bias occurs over a range of $|\hat{H}_i(\mathbf{u})|$ values.

Both the range and the magnitude of the bias are affected by n. Equations (4.3) and (4.59) show that for a fixed H_{max}, $\beta/|\hat{H}_i(\mathbf{u})|^{n+1}$ decreases as n increases, which results in a narrower range of $\hat{H}_i(\mathbf{u})$ associated with the bias. In other words, the range of $|\hat{H}_i(\mathbf{u})|$ affected by the bias becomes narrower as n increases. More components of $|\hat{H}_i(\mathbf{u})|$ contribute to the "ideal" range which resulting $|W_{\hat{H}_i}(\mathbf{u})|$ to approximate to the inverse filter. The magnitude of the bias, i.e. the difference between $|W_{\hat{H}_i}(\mathbf{u})|$ and $|I_{\hat{H}_i}(\mathbf{u})|$, gets smaller as n increases. The behaviour of $|W_{\hat{H}_i}(\mathbf{u})|$ compared with $|I_{\hat{H}_i}(\mathbf{u})|$ is demonstrated in Figure 4.12.

In the ineffective part of $\hat{H}_i(\mathbf{u})$ ($|\hat{H}_i(\mathbf{u})| < H_{max}$), the inverse filter is very susceptible to the contamination $C(\mathbf{u})$ due to the small values of $|\hat{H}_i(\mathbf{u})|$ in this range. The existence of $\beta/|\hat{H}_i(\mathbf{u})|^{n+1}$ enables the magnitude of the Wiener-like filter to be reduced suitably and therefore overcomes the shortcomings of the inverse filter. As n increases, $\beta/|\hat{H}_i(\mathbf{u})|^{n+1}$ increases (see (4.59)), and the suppression of $|W_{\hat{H}_i}(\mathbf{u})|$ is more effective as shown in Figure 4.12. Thus increasing n results in less influence of the contamination

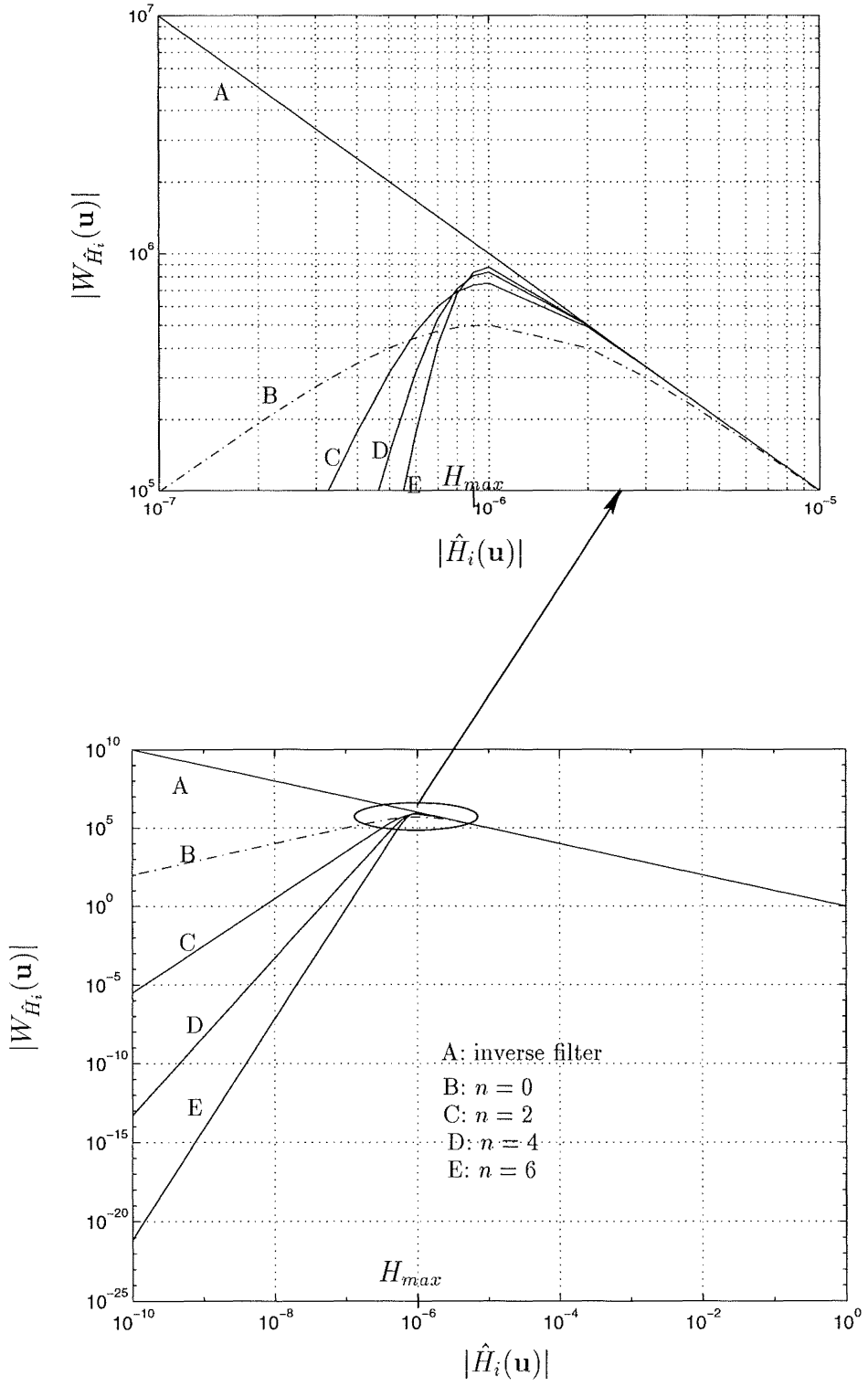


Figure 4.12 Demonstration of the effect of n on the relationship between the magnitude of the Wiener-like filter $W_{\hat{H}_i}(\mathbf{u})$ and $|\hat{H}_i(\mathbf{u})|$, both axes are logarithmic scaled; the cut-off spectral magnitude H_{max} is fixed in all cases of $n = 0, 2, 4$ and 6, and β is determined by H_{max} and n from (4.52).

of $G(\mathbf{u})$ in the new estimate $|F_i(\mathbf{u})|$.

The above analysis of the effect n on performance of the Wiener-like filter shows the advantage of selecting large n. However when n becomes large, although the bias is reduced to minimum for $|\hat{H}_i(\mathbf{u})| \geq H_{max}$ and the suppression to $|W_{\hat{H}_i}(\mathbf{u})|$ is most effective for $|\hat{H}_i(\mathbf{u})| < H_{max}$, a sharp transition of $|W_{\hat{H}_i}(\mathbf{u})|$ occurs between the two regions and results in artificial ringing in the restoration. It is therefore important to balance both the advantages and disadvantages when selecting the value of n in (4.3).

Experience in applying the Davey Algorithm with both $n = 0$ and $n = 2$ to deconvolve a number of contaminated blurred images indicated that the quality of the restorations from both values of n was very similar. However, in the majority of cases, the true error values of the images were slightly smaller for $n = 2$ than for $n = 0$.

4.2.4 Suggestion by Seldin and Fienup (1990)

Seldin and Fienup (1990) comment on Davey's Wiener-like filter (4.3), rewritten here for convenience with $n = 2$:

$$W_{\hat{H}_i}(\mathbf{u}) = \frac{\hat{H}_i^*(\mathbf{u})}{|\hat{H}_i(\mathbf{u})|^2 + \beta / |\hat{H}_i(\mathbf{u})|^2}. \tag{4.60}$$

Seldin and Fienup (1990) suggest that the term $\beta/|\hat{H}_i(\mathbf{u})|^2$ in (4.60) should be exchanged for $\beta/|\hat{F}_i(\mathbf{u})|^2$ (refer to the comments following Reference 6 of their paper). They base their argument on the Wiener-Helstrom filter (Helstrom, 1967), described by

$$WH(\mathbf{u}) = \frac{H^*(\mathbf{u})}{|H(\mathbf{u})|^2 + < |C(\mathbf{u})|^2 > / < |F(\mathbf{u})|^2 >}, \tag{4.61}$$

where $< |C(\mathbf{u})|^2 >$ and $< |F(\mathbf{u})|^2 >$ are the ensemble-averaged power spectra of the contamination $c(\mathbf{x})$ and the true image $f(\mathbf{x})$ respectively. To obtain (4.61), it is assumed that $f(\mathbf{x})$ and $c(\mathbf{x})$ are uncorrelated, gaussian processes with zero means. The Wiener-Helstrom filter is said to be formed to minimize the mean squared error between the true image and its estimate.

Seldin and Fienup (1990) pointed out that although the images generally do not satisfy the statistical assumptions stated above, the filter is still effective. Therefore they deduced the Wiener-like filter for the iterative blind deconvolution from the Wiener-Helstrom filter (4.61) as,

$$WH_{\hat{H}_i}(\mathbf{u}) = \frac{\hat{H}_i^*(\mathbf{u})}{|\hat{H}_i(\mathbf{u})|^2 + \sigma^2 / |\hat{F}_i(\mathbf{u})|^2}, \tag{4.62}$$

where $\hat{H}_i(\mathbf{u})$ approximates $H(\mathbf{u})$, $|\hat{F}_i(\mathbf{u})|^2$ is the estimate of $< |F(\mathbf{u})|^2 >$ and the constant σ^2 approximates $< |C(\mathbf{u})|^2 >$ in (4.61), provided that $c(\mathbf{x})$ is a Gaussian random process. This filter is henceforth refered to as the "SF filter". Seldin and Fienup (1990)

have demonstrated the performance of the algorithm by invoking (4.62) for the phase retrieval problem, which is a special case of blind deconvolution.

As has been analyzed and demonstrated in §4.2.2, the Davey filter (4.60) performs as a modified inverse filter and is effective. Its filter constant β is an estimate of $\frac{a^4}{3}\left|\frac{C(\mathbf{u})}{F(\mathbf{u})}\right|^4$ for $n = 2$ as described in (4.58), and $|\hat{H}_i(\mathbf{u})|^2$ in $\beta/|\hat{H}_i(\mathbf{u})|^2$ is used to reduce the effect of the bias existing in the Davey filter in (4.60). This suggests that the Davey filter and the SF filter may represent two different filters.

To further investigate the difference between the two filters, it is instructive to compare their performance. Since all the participants $\hat{H}_i(\mathbf{u})$, $|\hat{F}_i(\mathbf{u})|^2$ and σ^2 in (4.62) could be very poor estimates, respectively, of $H(\mathbf{u})$, $|F(\mathbf{u})|^2$ and $|C(\mathbf{u})|^2$ in the Wiener-Helstrom filter (4.61), they usually do not satisfy the statistical assumptions required by the optimal property associated with the filter. It is therefore sensible to view the SF filter more generally as a modified inverse filter. The cut-off spectral magnitude for the SF filter, denoted $H_{SFimax}(\mathbf{u})$, can be calculated by taking partial derivative of $|WH_{\hat{H}_i}(\mathbf{u})|$ with respect to $|\hat{H}_i(\mathbf{u})|$ and equating it to zero. That is

$$H_{SFimax}(\mathbf{u}) = \frac{\sigma}{|\hat{F}_i(\mathbf{u})|}. \tag{4.63}$$

Comparing (4.63) with (4.52) shows that the cut-off spectral magnitudes for the SF filter is very different from the Davey filter. Due to the participation of $\hat{F}_i(\mathbf{u})$, $H_{SFimax}(\mathbf{u})$ for the SF filter is a function of frequency $\mathbf{u}$ and also varies from iteration-to-iteration as the estimate $\hat{F}_i(\mathbf{u})$. The proportions of $|\hat{H}_i(\mathbf{u})|$ treated as effective and ineffective vary accordingly. If σ is an appropriate estimate of $|C(\mathbf{u})|$, the algorithm may however still converge to its solution. In contrast, H_{max} for the Davey filter in (4.52) is a constant for all frequencies and iterations. The algorithm converges gradually through the iterations, provided H_{max} is a suitable choice, until most of the components of $|\hat{H}_i(\mathbf{u})|$ above H_{max} are estimated.

To compare the effect of the Davey filter and the SF filter on blind deconvolution, the Davey Algorithm, employing these two filters separately, was applyed to a blurred image contaminated with SNR of 30dB (shown in Figure 4.3(c)). Correct supports of $f(\mathbf{x})$ and $h(\mathbf{x})$ were selected as their support estimates. Restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$, shown in Figure 4.13, were obtained at iterations where the minimum of the true error of $f(\mathbf{x})$, $E_T[f_i]$, occurred over 100 iterations. The true error $E_T[f_i]$ corresponding to both cases are shown in Figure 4.14.

Inspection of Figures 4.13 and 4.14 shows that the restorations retrieved using the Davey filter and the SF filter are similar in visual quality, however the true error $E_T[f_i]$ from the Davey filter is slightly lower than that from the SF filter. A significant difference between the performance of the algorithms associated with the two filters is that the Davey filter shows relatively stable convergence (*cf.* Figure 4.14). Deciding when to terminate iterations using the SF filter could be very difficult. Such erratic

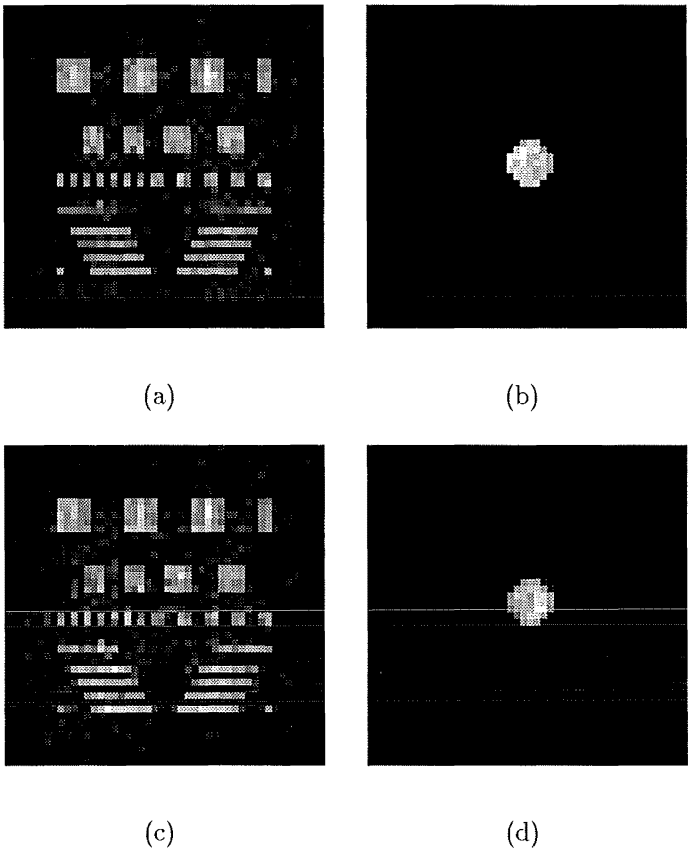


Figure 4.13 Comparison of using the Davey filter and the SF filter in blind deconvolution, SNR =30dB, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$, corresponding to Figures 4.3(a) and 4.3(b), from: (a) and (b) the Davey filter; (c) and (d) the SF filter.

behaviour of the algorithm when using the SF filter is expected since some components of $\hat{F}_i(\mathbf{u})$ could be quite poor estimates of that of $F(\mathbf{u})$, especially at high frequency region where the contaminated $G(\mathbf{u})$ does not contain useful information of $F(\mathbf{u})\,H(\mathbf{u})$. Consequently, the corresponding components of $H_{SFimax}(\mathbf{u})$ are poor estimates of their ideal quantity $|C(\mathbf{u})/F(\mathbf{u})|$, which causes the separation of the effective and ineffective parts of $\hat{H}_i(\mathbf{u})$ to be incompatible with usefulness of $G(\mathbf{u})$ in a manner which varies with $\hat{F}_i(\mathbf{u})$ as iterations proceed.

Note that such erratic behaviour of $E_T[f_i]$ was not shown when the algorithm was applied for the phase retrieval problem (Seldin and Fienup, 1990). This is because in that case, $\hat{F}_i(\mathbf{u}) = \hat{H}_i^*(\mathbf{u})$, and equation (4.62) becomes

$$WH_i(\mathbf{u}) = \frac{\hat{F}_i(\mathbf{u})}{|\hat{F}_i(\mathbf{u})|^2 + \sigma^2 / \,|\hat{F}_i(\mathbf{u})|^2}. \tag{4.64}$$

The cut-off spectral magnitude in this case, denoted F_{SFmax}, is

$$F_{SFmax} = (3\sigma^2)^{\frac{1}{4}}. \tag{4.65}$$

F_{SFmax} is a constant rather than a function of $\hat{F}_i(\mathbf{u})$. If F_{SFmax} is suitably chosen, it

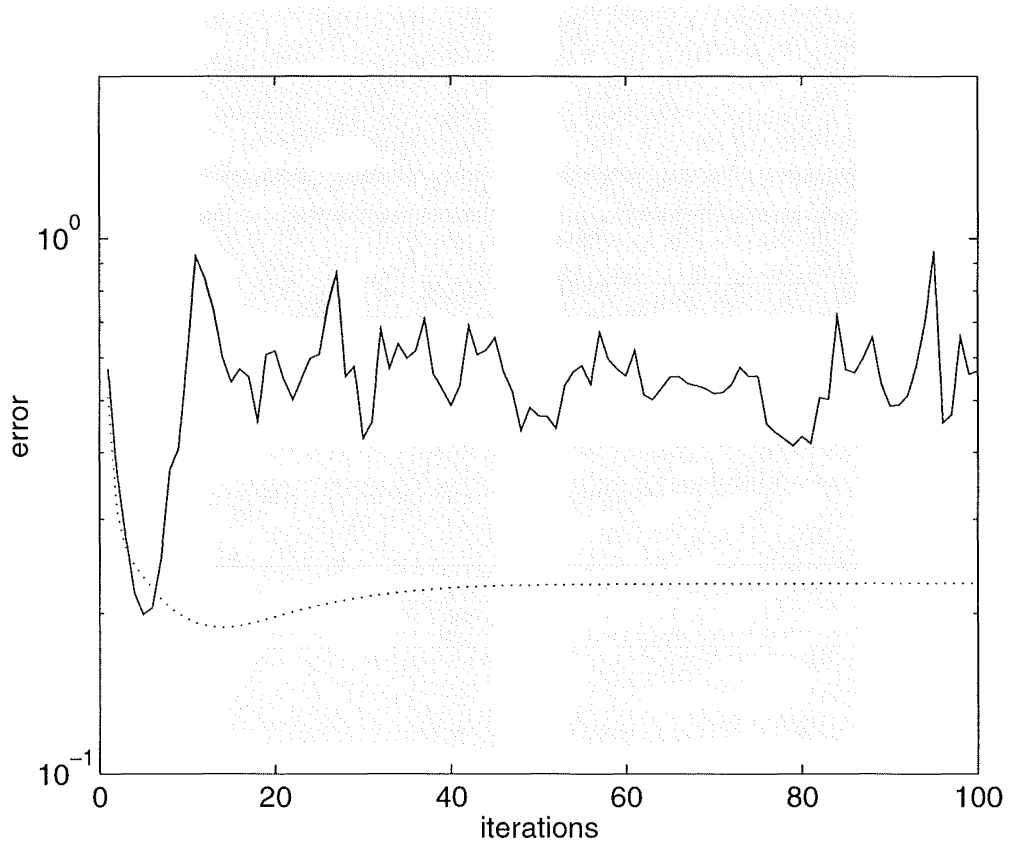


Figure 4.14 True errors of $f(\mathbf{x})$ corresponding to Figure 4.13. Dotted and continuous curves represent the Davey filter and the SF filter respectively.

can be compatible with the effectiveness of $G(\mathbf{u})$ throughout the iterations.

4.3 MONITORING THE CONVERGENCE

This section focuses on monitoring the performance of the Davey Algorithm. Section 4.3.1 investigates how well the performance of the algorithm is evaluated by some objective error measures. Based on this investigation, a method of determining the best support estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ is presented in §4.3.2. This result has already reported (Jiang and Bones, 1991).

4.3.1 The behaviour of objective error measures

The behaviour of two types of objective error measures, referred to as the image space error and the convolutional error, are investigated in terms of applying the algorithm using different sizes of support estimates and various values of the filter constant β, which are described in §4.3.1.1 and §4.3.1.2 respectively. This error behaviour is compared with

the true error, $E_T[f_i]$ (*cf.* (3.21)). The image space error, denoted $E_I[f_i]$ for $f_i(\mathbf{x})$, has been introduced in (3.22). $E_I[h_i]$ for $h_i(\mathbf{x})$ can be calculated in a similar manner. The convolutional error is defined as the normalized squared difference between the given blurred image $g(\mathbf{x})$ and the estimate of the convolution. Two kinds of convolutional error can be used: one, denoted $E_b[g_i]$, is calculated from the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ obtained before the image space constraints are applied. That is

$$E_b[g_i] = \frac{\int_{\mathbf{x}} |g(\mathbf{x}) - \varrho_b^i f_i(\mathbf{x}) \odot h_i(\mathbf{x})|^2 \, d\mathbf{x}}{\int_{\mathbf{x}} |g(\mathbf{x})|^2 \, d\mathbf{x}}, \tag{4.66}$$

where $\varrho_b^i = \text{Eng}[g(\mathbf{x})]/\text{Eng}[f_i(\mathbf{x}) \odot h_i(\mathbf{x})]$. The other, defined $E_a[g_i]$, is calculated after the constraints are enforced,

$$E_a[g_i] = \frac{\int_{\mathbf{x}} |g(\mathbf{x}) - \varrho_a^i \hat{f}_{i+1}(\mathbf{x}) \odot \hat{h}_i(\mathbf{x})|^2 \, d\mathbf{x}}{\int_{\mathbf{x}} |g(\mathbf{x})|^2 \, d\mathbf{x}}, \tag{4.67}$$

where $\varrho_a^i = \text{Eng}[g(\mathbf{x})]/\text{Eng}[\hat{f}_{i+1}(\mathbf{x}) \odot \hat{h}_i(\mathbf{x})]$. By means of ϱ_b^i and ϱ_a^i, the energies of the estimates of $g(\mathbf{x})$ in (4.66) and (4.67) respectively are normalized to the energy of $g(\mathbf{x})$. Both $E_b[g_i]$ and $E_a[g_i]$ indicate to what extent the convolution estimate differs from the given measurement. For simplicity, only one of the convolutional errors, $E_b[g_i]$, is investigated here.

4.3.1.1 Effect of support estimate

The behaviour of $E_b[g_i]$ is firstly examined when the algorithm is performed using different sizes of support estimates. The same examples shown in Figures 4.5 and 4.6 are used here. Figure 4.15 plots $E_b[g_i]$ corresponding to the restorations shown in Figure 4.5, where $f(\mathbf{x})$ is a bilevel image with 32×32 pixels in extent in each direction and $h(\mathbf{x})$ is a disc with diameter of 7 pixels, $g(\mathbf{x})$ is contaminated by SNR of 30dB. Figure 4.16 shows $E_b[g_i]$ corresponding to Figure 4.6, where $f(\mathbf{x})$ is a bilevel image with 16×16 pixels in extent, $h(\mathbf{x})$ is formed by pseudo-random values uniformly distributed within a circular region with diameter of 7 pixels, and $g(\mathbf{x})$ is contaminated by SNR of 20dB. In each of the two figures, three curves are shown, corresponding to three different combination of the sizes of support estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$, $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$, namely:

case a) $M_f(\mathbf{x}) > B_f(\mathbf{x})$ and $M_h(\mathbf{x}) < B_h(\mathbf{x})$;

case b) $M_f(\mathbf{x}) = B_f(\mathbf{x})$ and $M_h(\mathbf{x}) = B_h(\mathbf{x})$;

case c) $M_f(\mathbf{x}) < B_f(\mathbf{x})$ and $M_h(\mathbf{x}) > B_h(\mathbf{x})$.

In all cases, the image-box of $g(\mathbf{x})$ is assumed to be known precisely and both dimensions of the support estimates are altered equally.

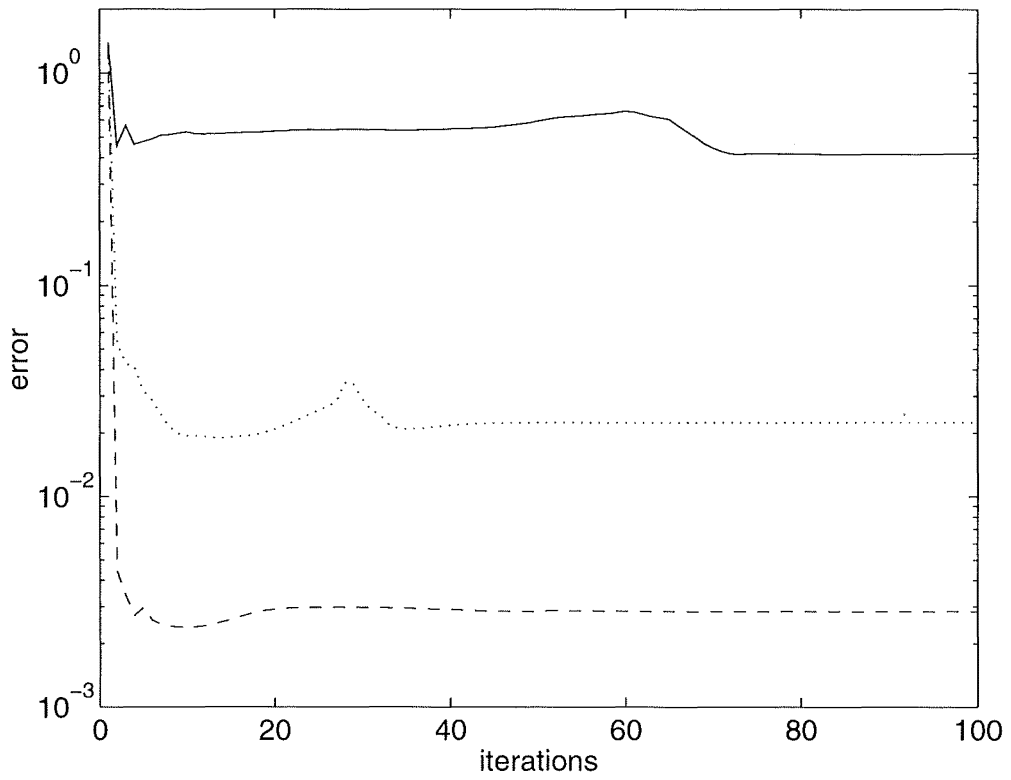


Figure 4.15 Illustrating the effect of using different sizes of support estimates on the behaviour of $E_b[g_i]$, dashed curve: case a) $M_f(\mathbf{x}) = 36 \times 36$ pixels, $M_h(\mathbf{x}) = 3 \times 3$ pixels; dotted curve: case b) $M_f(\mathbf{x}) = 32 \times 32$ pixels, $M_h(\mathbf{x}) = 7 \times 7$ pixels; continuous curve: case c) $M_f(\mathbf{x}) = 28 \times 28$ pixels, $M_h(\mathbf{x}) = 11 \times 11$ pixels; these curves correspond to restorations shown in Figure 4.5.

Inspection of the error curves shown in Figures 4.15 and 4.16 reveals that the lowest value of $E_b[g_i]$ among the tested support estimates often results from overestimating $M_f(\mathbf{x})$ and underestimating $M_h(\mathbf{x})$ rather than from the best estimates of both supports (*cf.* dotted curve in Figure 4.15 and curve "C" in Figure 4.16). However, the restorations from the former case are still quite blurred for $f_i(\mathbf{x})$ (*cf.* Figures 4.5(a), 4.6(e) and 4.6(f)) and truncated or unrecovered for $h_i(\mathbf{x})$ (*cf.* Figures 4.5(d), 4.6(i) and 4.6(j)), while the restorations from the latter show a more convincing deconvolution (*cf.* Figures 4.5(b), 4.5(e) 4.6(g) and 4.6(k)). This observation suggests that the convolutional error appears unable to reliably detect the best initial support estimates.

The effect of different support estimates on the image space errors is now investigated. To observe both the individual and joint effects of the two estimates $f_i(\mathbf{x})$ and $h_i(\mathbf{x})$ on the image space errors, three error curves (representing $E_I[f_i]$, $E_I[h_i]$ and the summation of $E_I[f_i]$ and $E_I[h_i]$) corresponding to the examples shown in Figure 4.5 are plotted in Figure 4.17 and those corresponding to the examples of Figure 4.6 are plotted

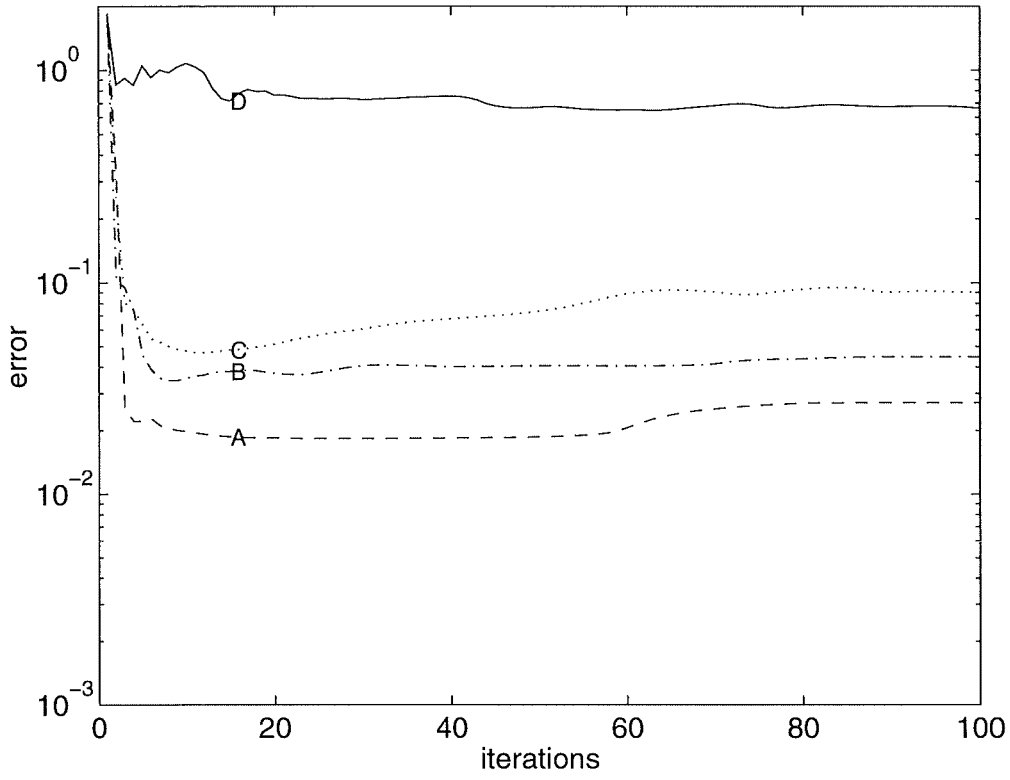


Figure 4.16 Second example of the effect of support estimates on $E_b[g_i]$, curve A: case a) $M_f(\mathbf{x}) = 20 \times 20$ pixels, $M_h(\mathbf{x}) = 3 \times 3$ pixels; curve B: case a) $M_f(\mathbf{x}) = 18 \times 18$ pixels, $M_h(\mathbf{x}) = 5 \times 5$ pixels; curve C: case b) $M_f(\mathbf{x}) = 16 \times 16$ pixels, $M_h(\mathbf{x}) = 7 \times 7$ pixels; curve D: case c) $M_f(\mathbf{x}) = 14 \times 14$ pixels, $M_h(\mathbf{x}) = 9 \times 9$ pixels, these curves correspond to restorations depicted in Figure 4.6.

in Figure 4.18.

Figures 4.17(a) and 4.18(a) show that $E_I[f_i]$ appears to depend substantially on the support estimate chosen for $f(\mathbf{x})$; larger support estimate $M_f(\mathbf{x})$ often leads to a lower value of the image space error, despite the best restoration being obtained with correct support at each dimensions (*cf.* dotted curves in Figure 4.17 and curve "C" Figure 4.18). Clearly using $E_I[f_i]$ to choose the best support is inappropriate.

However, the behaviour of $E_I[h_i]$ is significantly different from that of $E_I[f_i]$. Figures 4.17(b) and 4.18(b) show that, for the best support estimate (*cf.* case b)), $E_I[h_i]$ is usually lower than for other cases after a small number (say 20) of iterations. This indicates that $E_I[h_i]$ is much less affected by the size of the related support estimate than $E_I[f_i]$ is. Therefore, $E_I[h_i]$ appears to be capable of detecting the best support estimate.

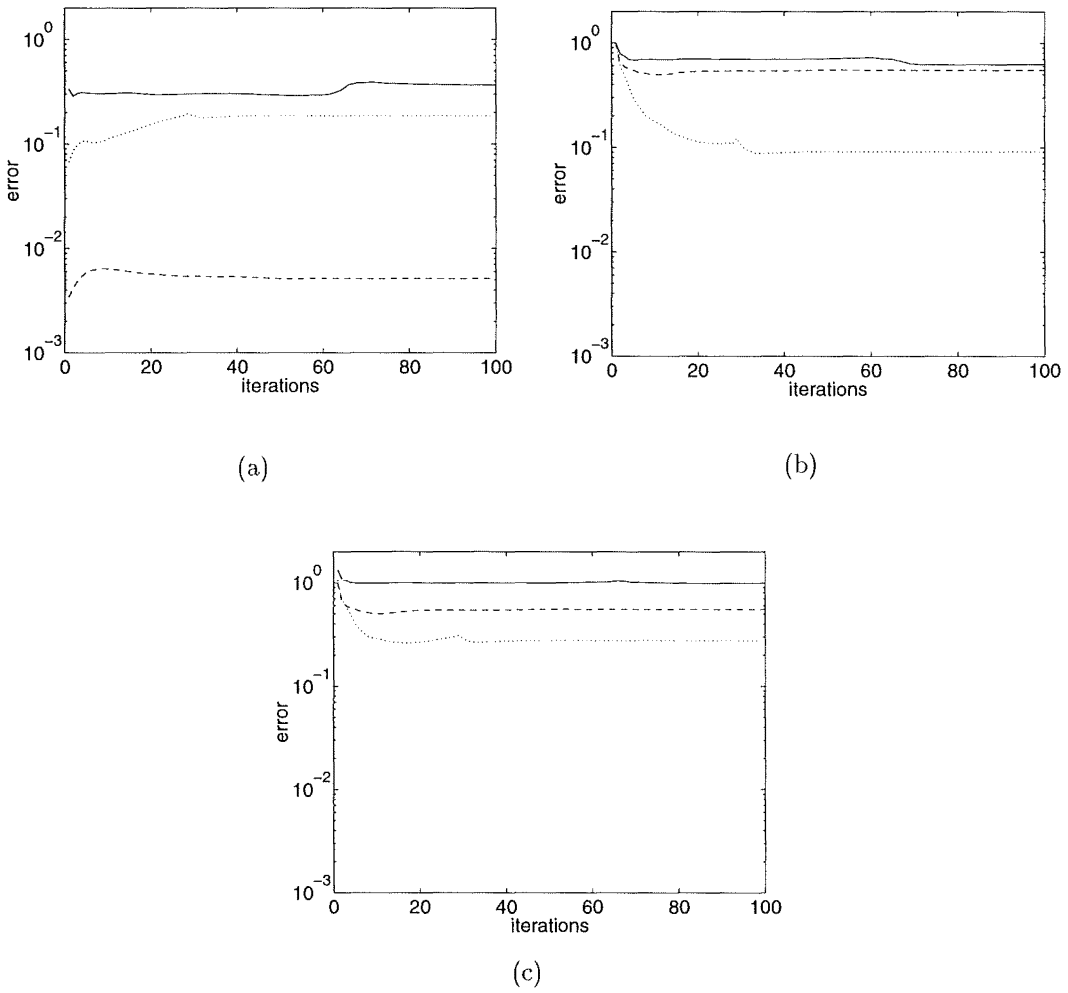


Figure 4.17 Demonstrating the effect of different support sizes on the image space error, corresponding to Figure 4.5: (a) $E_I[f_i]$; (b) $E_I[h_i]$; (c) $E_I[f_i] + E_I[h_i]$; dashed curve: case a) $M_f(\mathbf{x}) = 36 \times 36$ pixels, $M_h(\mathbf{x}) = 3 \times 3$ pixels; dotted curve: case b) $M_f(\mathbf{x}) = 32 \times 32$ pixels, $M_h(\mathbf{x}) = 7 \times 7$ pixels; continuous curve: case c) $M_f(\mathbf{x}) = 28 \times 28$ pixels, $M_h(\mathbf{x}) = 11 \times 11$ pixels.

The sum of the image space errors $E_I[f_i] + E_I[h_i]$ achieves its lowest value with the best estimates of $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$ in Figure 4.17(c). In contrast, in Figure 4.18(c) the pair of overestimated $M_f(\mathbf{x})$ and underestimated $M_h(\mathbf{x})$ results in the lowest value of the sum of $E_I[f_i]$ and $E_I[h_i]$.

The above are typical results from around a dozen tests performed. The investigation shows that: $E_b[g_i]$ and $E_I[f_i]$ are basically not useful for determining the best support estimates; $E_I[f_i]+E_I[h_i]$ is sometimes useful amongst the tested data; and $E_I[h_i]$ appears to be useful in detecting the best support estimates.

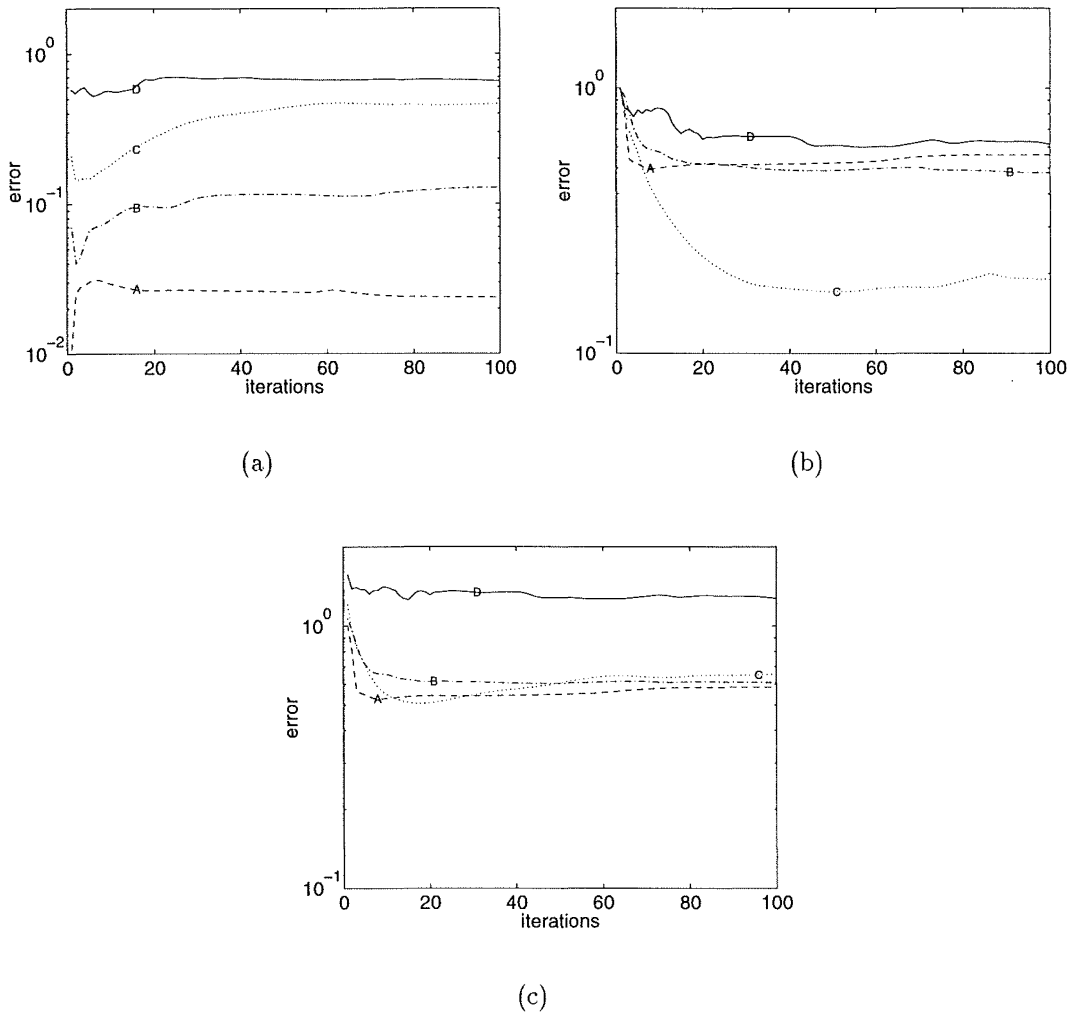


Figure 4.18 Second example of the effect of support sizes on the image space error, corresponding to restorations in Figure 4.6: (a) $E_I[f_i]$; (b) $E_I[h_i]$; (c) $E_I[f_i]+E_I[h_i]$, curve A: case a) $M_f(\mathbf{x})=20\times 20$ pixels, $M_h(\mathbf{x})=3\times 3$ pixels; curve B: case a) $M_f(\mathbf{x})=18\times 18$ pixels, $M_h(\mathbf{x})=5\times 5$ pixels; curve C: case b) $M_f(\mathbf{x})=16\times 16$ pixels, $M_h(\mathbf{x})=7\times 7$ pixels; curve D: case c) $M_f(\mathbf{x})=14\times 14$ pixels, $M_h(\mathbf{x})=9\times 9$ pixels.

4.3.1.2 Effect of filter constant β

The behaviour of the convolutional error $E_b[g_i]$ and the image space errors of $E_I[f_i]$, $E_I[h_i]$ and $E_I[f_i]+E_I[h_i]$ are now investigated when the algorithm is applied with different values of the filter constant β. For convenience of comparison, the true error $E_T[f_i]$ is shown as well. The blind deconvolution example shown in Figure 4.3 is used here. The contamination level of the blurred image is SNR of 30dB. Figure 4.19 displays the two types of objective error measures, the image space errors $E_I[f_i]$, $E_I[h_i]$ and $E_I[f_i]+E_I[h_i]$, and the convolutional error $E_b[g_i]$, and the true error $E_T[f_i]$ corresponding to Figure 4.3. Inspection of the error curves in Figure 4.19 reveals that $E_I[f_i]$ neither corresponds well with $E_T[f_i]$ nor be compatible with visual

observation of the restorations shown in Figure 4.3. Increasing value of β tends to result in smaller values of $E_I[f_i]$, while the quality of the restorations are not necessarily improved. Although the minima of both $E_I[f_i]+E_I[h_i]$ and $E_b[g_i]$ in Figure 4.19 result from a different value of β ($\beta = 5\times10^{-7}$) compared with the minima of $E_T[f_i]$ ($\beta = 5\times10^{-8}$), inspection of the corresponding restorations in Figures 4.3(b) and 4.3(c) shows that the restoration due to $\beta = 5 \times 10^{-7}$ is comparable to that due to $\beta = 5 \times 10^{-8}$. Therefore, the lowest trajectories for $E_I[f_i] + E_I[h_i]$ and $E_b[g_i]$ in Figure 4.19 correspond to a suitable choice of β.

In summary, the behaviour of the objective error measures resulting from different values of β shows that the minimum of the convolutional error $E_b[g_i]$ tends to coincide with the best restorations and indicates a sensible choice of β. The minimum $E_b[g_i]$ also coincides reasonably well with the minimum of $E_T[f_i]$, a feature which is valuable for monitoring the performance of the algorithm, for example, determining when to stop the iterative process or deciding when and how to dynamically change β (see §5.2.2.4).

Further investigation of the effect of β on $E_b[g_i]$ has been carried out when $g(\mathbf{x})$ was contaminated to various levels. Results indicated that the correlation between $E_b[g_i]$ and $E_T[f_i]$ existed until the contamination level of $g(\mathbf{x})$ became as high as SNR = 20dB.

4.3.2 Estimating the best supports

Investigation of the objective error measures presented in §4.3.1 shows that the behaviour of the image space errors $E_I[h_i]$ and $E_I[f_i]$ is very different for various support estimates. $E_I[f_i]$ tends to be overly affected by the estimated support sizes, while $E_I[h_i]$ does not. One of the reasons for this behaviour of $E_I[f_i]$ may be explained on the basis of the size of the set of pixels violating the support constraint. Overestimating $M_f(\mathbf{x})$ lowers the number of pixels of $f_i(\mathbf{x})$ which violate the support constraint, i.e. the size of the set Ω_i^f is reduced compared to the size of Ω_i^f for correct $M_f(\mathbf{x})$. The numerator of the expression for $E_I[f_i]$ reflects the size of Ω_i^f. Since it is most improbable that a restored pixel has exactly zero value, it is likely that all members of Ω_i^f contribute to $E_I[f_i]$. The smaller set of Ω_i^f resulting from the overestimated $M_f(\mathbf{x})$ therefore results in a lower value of $E_I[f_i]$ compared to that from the correct $M_f(\mathbf{x})$.

The effect of support size on $E_I[h_i]$ is quite different. The minimum of $E_I[h_i]$ for a range of support estimates corresponds well with the best support estimates. This may due to the fact that $S_h(\mathbf{x})$ is usually much smaller than the corresponding $S_f(\mathbf{x})$ in situations of practical interest. If both $S_h(\mathbf{x})$ and $S_f(\mathbf{x})$ are overestimated by an equal number of pixels in each extent, relative changes in Ω_i^h due to changes in $M_h(\mathbf{x})$ are much smaller than changes in Ω_i^f due to equal changes in the extents of $M_f(\mathbf{x})$. Therefore, the effect of the overestimated $M_h(\mathbf{x})$ on $E_I[h_i]$ is much less than that of the overestimated $M_f(\mathbf{x})$ on $E_I[f_i]$. It seems likely that $E_I[h_i]$ is more sensitive to the actual error in restoration, rather than to the size of the set of pixels over which it is calculated.

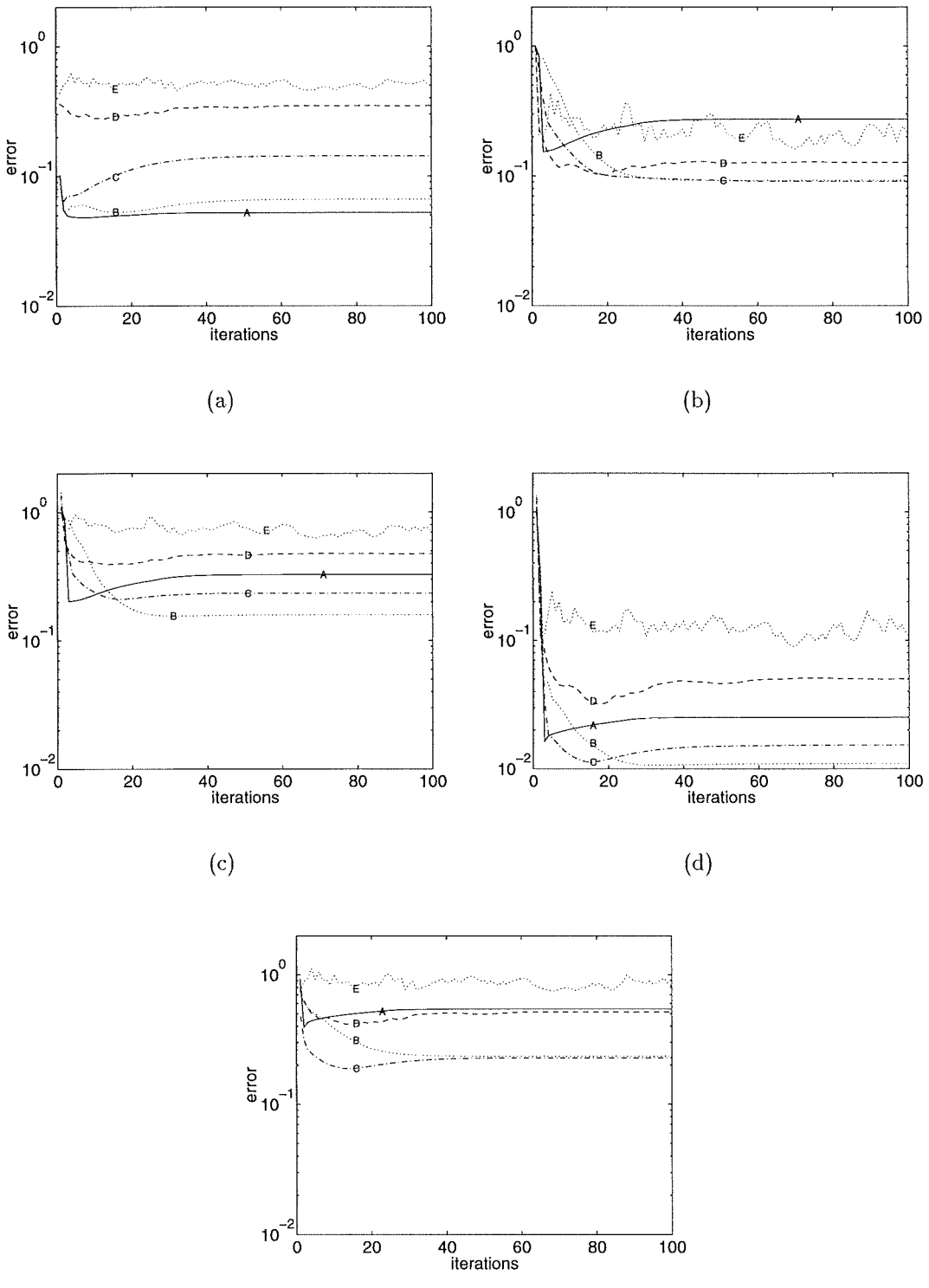


Figure 4.19 Illustrating the effect of using different values of the filter constant β on the various error measures, curve A: $\beta = 5 \times 10^{-6}$, curve B: $\beta = 5 \times 10^{-7}$, curve C: $\beta = 5 \times 10^{-8}$, curve D: $\beta = 5 \times 10^{-9}$, curve E: $\beta = 5 \times 10^{-10}$, these error plots correspond to the restorations shown in Figure 4.3. (a) $E_I[f_i]$; (b) $E_I[h_i]$; (c) $E_I[f_i] + E_I[h_i]$; (d) $E_b[g_i]$; and (e) $E_T[f_i]$.

For example, the additional pixels on each side of $M_h(\mathbf{x})$ or $M_f(\mathbf{x})$ number four in the example of Figure 4.17, so that the number of pixels of $h_i(\mathbf{x})$ which are missed out by the support constraint is 72, only 1.8% of the total number of pixels (64×64). However, the number of pixels missed out in $f_i(\mathbf{x})$ is 272, 6.6% of the total. It follows that the contribution of the pixels left out is much less in $h_i(\mathbf{x})$ than in $f_i(\mathbf{x})$.

The sum $E_I[f_i]+E_I[h_i]$ is usually significantly influenced by $E_I[h_i]$. If $S_h(\mathbf{x})$ is much smaller than $S_f(\mathbf{x})$ (as in the example corresponding to Figure 4.17 where $f(\mathbf{x})$ is 32×32 pixels in extent in each direction and $S_h(\mathbf{x})$ has diameter of 7 pixels), many more pixels of $h_i(\mathbf{x})$ violate the support constraint compared to number of violating pixels of $f_i(\mathbf{x})$. Therefore, the corresponding $E_I[h_i]$ is much larger than $E_I[f_i]$. The sum $E_I[f_i]+E_I[h_i]$ is then dominated by $E_I[h_i]$ and appears to indicate the best support estimates. If the relative sizes of $S_h(\mathbf{x})$ compared with $S_f(\mathbf{x})$ in each extent is comparable (such as those relating to Figure 4.18 where $f(\mathbf{x})$ is surrounded within a region of 16×16 pixels in extent and $S_h(\mathbf{x})$ is a disc with diameter of 7 pixels), the value of $E_I[h_i]$ lies in a similar range to that of $E_I[f_i]$. The low value of $E_I[f_i]$ resulted from the overestimated $M_f(\mathbf{x})$ lowers the value of $E_I[f_i]+E_I[h_i]$. In this case $E_I[f_i]+E_I[h_i]$ fails to detect the best support estimate.

The results of this initial investigation suggest that $E_I[h_i]$ may be a useful indicator of the best choice of the estimate of $S_h(\mathbf{x})$. However, the experiment is based on a good choice of the filter constant β. There is practical importance in further investigating the behaviour of $E_I[h_i]$ in terms of a wider range of values of β, a wider range of contamination levels, and larger sizes of $S_h(\mathbf{x})$.

The example presented in Figure 4.5, for which the convolutional and image space errors are shown in Figures 4.15 and 4.17, is used here to further investigate the effect of support estimates on $E_I[h_i]$. $E_I[h_i]$ is produced when the algorithm is applied using different support estimates, $g(\mathbf{x})$ is contaminated to within the range of levels with SNR $= 40, 30, 20$ and 10dB and the values of β is selected ranging from 5×10^{-7} to 5×10^{-9}. The resulting minima of $E_I[h_i]$ within 100 iterations are tabulated in 4.1.

Table 4.1 shows that for the tested ranges of SNR and β, all the lower values of $E_I[h_i]$ correspond to the best support estimate $M_h(\mathbf{x})$ (i.e. 7×7 pixels).

The effect of the larger support estimate $M_h(\mathbf{x})$ on $E_I[h_i]$ is further investigated. The true image $f(\mathbf{x})$ (see Figure 4.20(a)) is a positive bilevel image with 32×32 pixels in extent in each coordinate direction, the psf $h(\mathbf{x})$ (see Figure 4.20(b)) features a "J" shape with 16×16 pixels in extent, and is exactly the same as the $f(\mathbf{x})$ shown in Figure 4.6(a). The blurred image $g(\mathbf{x})$ (as shown in Figure 4.20(c)) is formed by the convolution of $f(\mathbf{x})$ and $h(\mathbf{x})$ which is contaminated with SNR of 30dB. Blind deconvolution using five pairs of support estimates, respectively, ranging from 40×40 to 24×24 pixels for $M_f(\mathbf{x})$ and from 8×8 to 24×24 pixels for $M_h(\mathbf{x})$, has been attempted. Restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ are presented in Figure 4.21 and the corresponding image space errors

SNR (dB)	Filter constant β	Support estimates $M_f(\mathbf{x})$, $M_h(\mathbf{x})$ (pixels)			
		24×24	28×28	32×32	36×36
		15×15	11×11	7×7	3×3
40dB	5×10^{-7}	0.654	0.554	0.067	0.184
	5×10^{-8}	0.642	0.660	0.076	0.383
	5×10^{-9}	0.719	0.642	0.035	0.216
30dB	5×10^{-7}	0.663	0.552	0.078	0.207
	5×10^{-8}	0.687	0.631	0.088	0.496
	5×10^{-9}	0.707	0.621	0.094	0.309
20dB	5×10^{-7}	0.768	0.604	0.180	0.368
	5×10^{-8}	0.836	0.687	0.258	0.671
	5×10^{-9}	0.836	0.738	0.344	0.602
10dB	5×10^{-7}	0.856	0.786	0.541	0.626
	5×10^{-8}	0.895	0.836	0.616	0.806
	5×10^{-9}	0.910	0.873	0.745	0.858

Table 4.1 The image space error $E_I[h_i]$ resulted from different values of β, a range of SNR levels and different sizes of support estimates $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$. The related $f(\mathbf{x})$, $h(\mathbf{x})$, $g(\mathbf{x})$ and $\hat{f}_0(\mathbf{x})$ correspond to those shown in Figure 4.5, where $f(\mathbf{x})$ is 32×32 pixels and $h(\mathbf{x})$ is circular with a diameter of 7 pixels.

$E_I[h_i]$ are shown in Figure 4.22. Inspection of Figures 4.21 and 4.22 reveals when the extent in each direction of $S_h(\mathbf{x})$ reaches as much as half of that of $S_f(\mathbf{x})$, the minimum of $E_I[h_i]$ in Figure 4.22 still corresponds to the correct choice of $S_h(\mathbf{x})$. Therefore $E_I[h_i]$ remains capable of indicating the best support estimate.

Note when the same $h(\mathbf{x})$ in Figure 4.20(b) is used as $f(\mathbf{x})$ in Figure 4.6(a), the lowest image space error $E_I[f_i]$ (Figure 4.18(a)) does not match with the correct $M_f(\mathbf{x})$. This suggests that the effectiveness of using $E_I[h_i]$ to estimate $S_h(\mathbf{x})$ relates to both the relative size of $S_f(\mathbf{x})$ compared with $S_h(\mathbf{x})$ in each extent, and the relative size of $S_h(\mathbf{x})$ compared with the total size of $h_i(\mathbf{x})$ (32×32 pixels in Figure 4.18 and 64×64 pixels in Figure 4.22) in which $S_h(\mathbf{x})$ is embedded.

The investigation of the effect of different support sizes on the two objective error measures shows that the image space error of $h_i(\mathbf{x})$, $E_I[h_i]$, has been found consistently to indicate the best support dimensions when the algorithm is applied to a contaminated blurred image $g(\mathbf{x})$ with a wide range of contamination levels and a range of filter constant values. It is therefore demonstrated that $E_I[h_i]$ can be used to estimate the

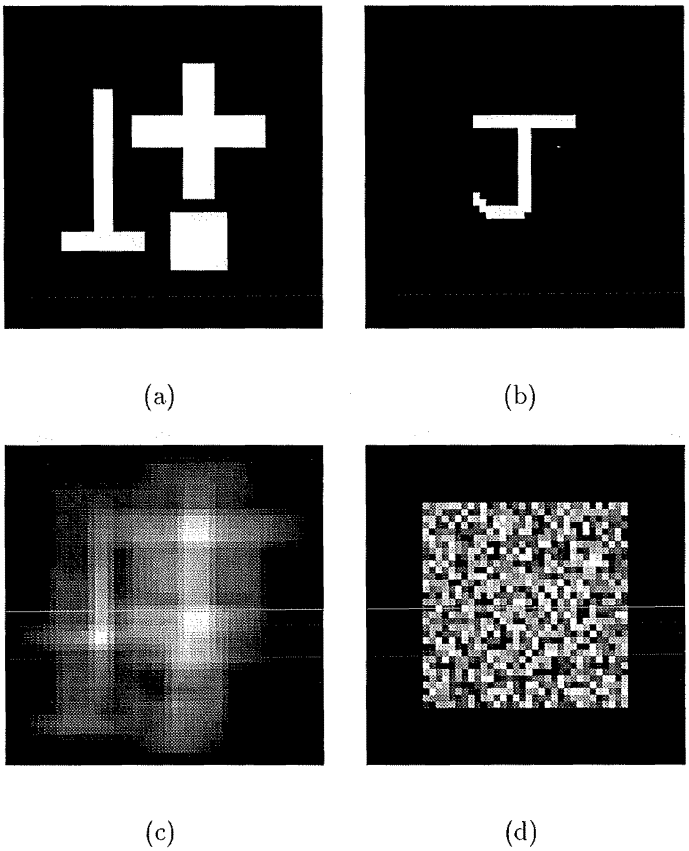


Figure 4.20 (a) A real and positive image $f(\mathbf{x})$; (b) psf $h(\mathbf{x})$; (c) blurred image $g(\mathbf{x}) = f(\mathbf{x}) \odot h(\mathbf{x}) + c(\mathbf{x})$ with SNR of 30dB; and (d) initial estimate of $f(\mathbf{x})$, $\hat{f}_0(\mathbf{x})$.

supports of $f(\mathbf{x})$ and $h(\mathbf{x})$ in practice. The support estimate is one of the key parameters to the successful convergence of the algorithm.

The determination of the best support estimate are described in the following:

(i) Determine $M_g(\mathbf{x})$ from the image-box of $g(\mathbf{x})$ through thresholding $g(\mathbf{x})$ appropriately.

(ii) Make an initial estimate, $M_h(\mathbf{x})$, of the size of the rectangular support of $S_h(\mathbf{x})$.

(iii) Calculate $M_f(\mathbf{x})$ from $M_g(\mathbf{x})$ and $M_h(\mathbf{x})$ by invocation of the convolution theorem.

(iv) Select the filter constant β (say 10^{-8}) and starting image $\hat{f}_0(\mathbf{x})$, and run the algorithm.

(v) Calculate $E_I[h_i]$ at each iteration of the algorithm.

(vi) Increase or decrease the extent of each side of $M_h(\mathbf{x})$ (by 2 pixels, say) and form a new estimate $M_h(\mathbf{x})$.

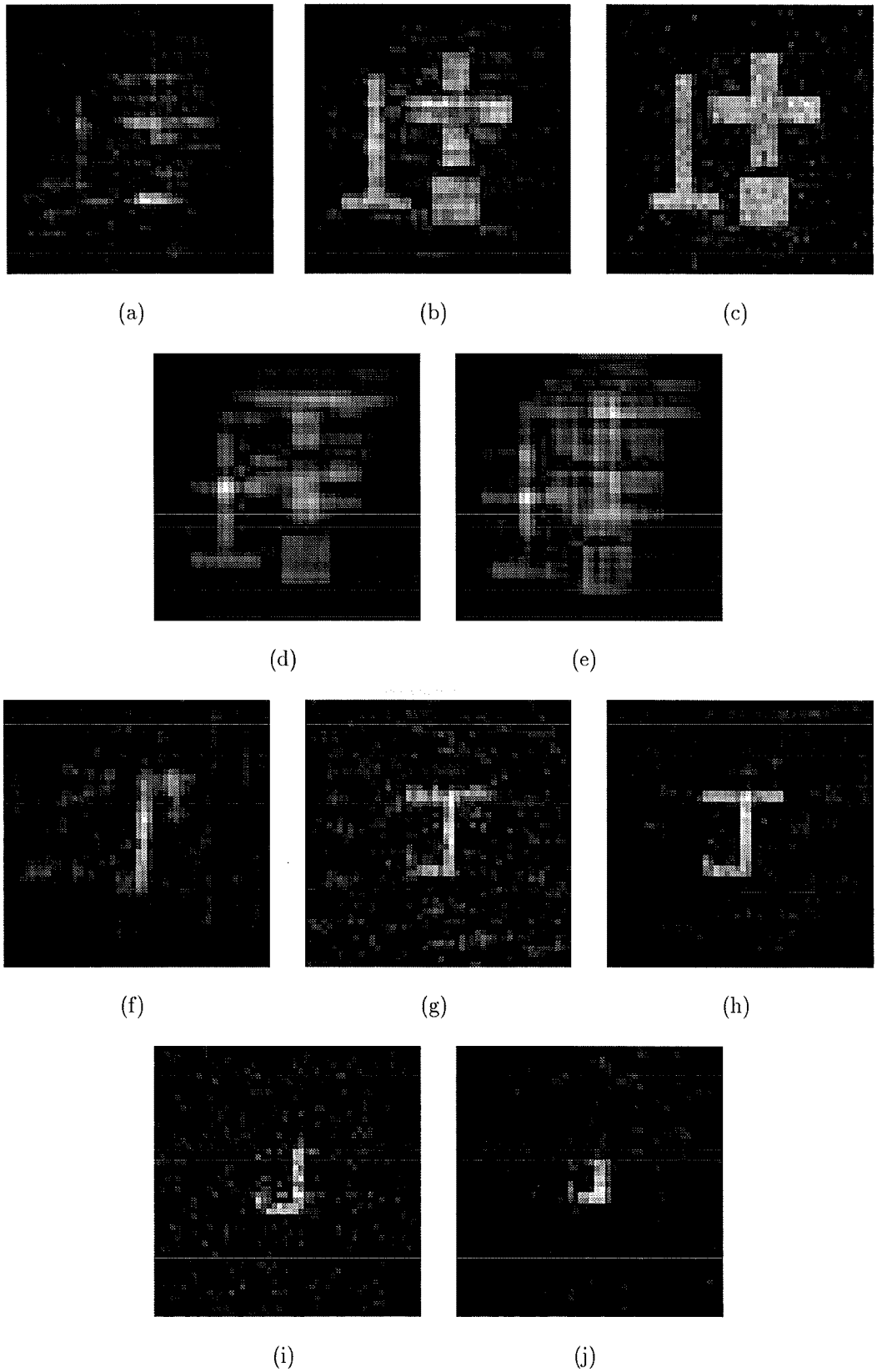


Figure 4.21 An example of demonstrating the effect of large $S_h(\mathbf{x})$ on the image space error $E_I[h_i]$, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ obtained when the Davey Algorithm was applied to $g(\mathbf{x})$ shown in Figure 4.20(c) using different support estimates: (a) and (f) case a) $M_f(\mathbf{x}) = 40 \times 40$ pixels, $M_h(\mathbf{x}) = 8 \times 8$ pixels; (b) and (g) case a) $M_f(\mathbf{x}) = 36 \times 36$ pixels, $M_h(\mathbf{x}) = 12 \times 12$ pixels; (c) and (h) case b) $M_f(\mathbf{x}) = 32 \times 32$ pixels, $M_h(\mathbf{x}) = 16 \times 16$ pixels ($M_f(\mathbf{x}) = B_f(\mathbf{x})$ and $M_h(\mathbf{x}) = B_h(\mathbf{x})$); (d) and (i) case c) $M_f(\mathbf{x}) = 28 \times 28$ pixels, $M_h(\mathbf{x}) = 20 \times 20$ pixels; (e) and (j) case c) $M_f(\mathbf{x}) = 24 \times 24$ pixels, $M_h(\mathbf{x}) = 24 \times 24$ pixels.

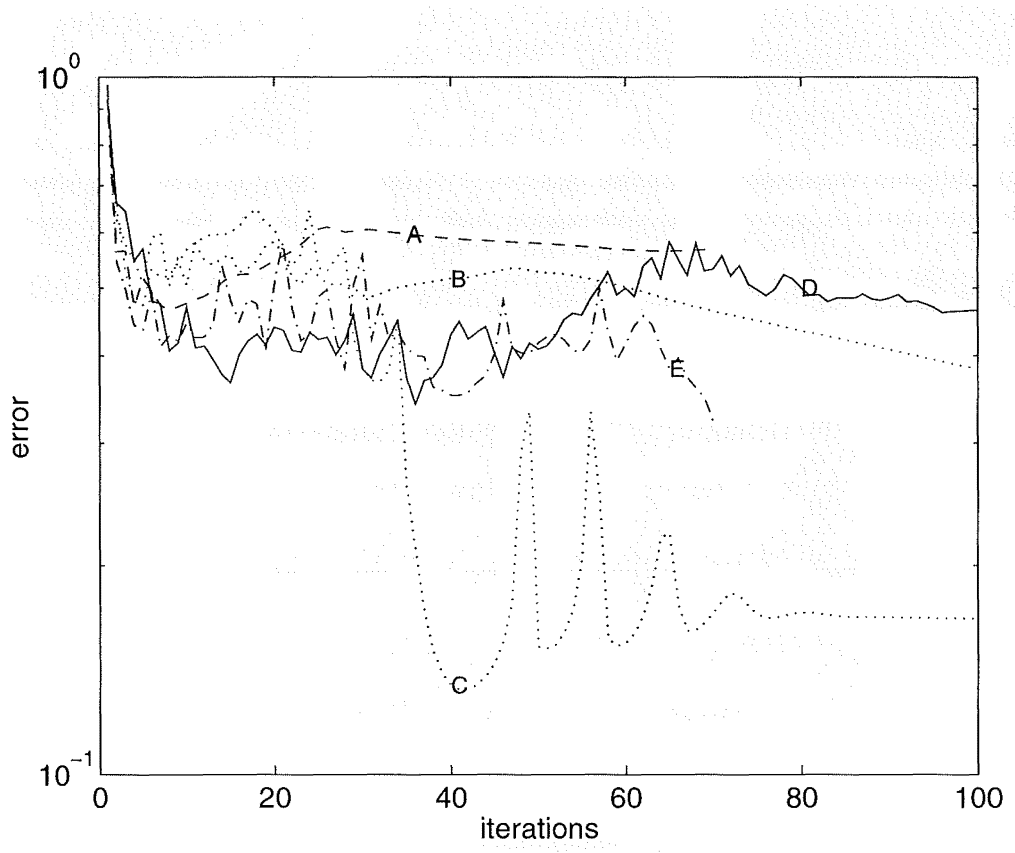


Figure 4.22 The image space error $E_I[h_i]$ corresponding to Figure 4.21. Curve A: case a) $M_f(\mathbf{x}) = 40 \times 40$ pixels, $M_h(\mathbf{x}) = 8 \times 8$ pixels; curve B: case a) $M_f(\mathbf{x}) = 36 \times 36$ pixels, $M_h(\mathbf{x}) = 12 \times 12$ pixels; curve C: case b) $M_f(\mathbf{x}) = 32 \times 32$ pixels, $M_h(\mathbf{x}) = 16 \times 16$ pixels ($M_f(\mathbf{x}) = B_f(\mathbf{x})$ and $M_h(\mathbf{x}) = B_h(\mathbf{x})$); curve D: case c) $M_f(\mathbf{x}) = 28 \times 28$ pixels, $M_h(\mathbf{x}) = 20 \times 20$ pixels; curve E: case c) $M_f(\mathbf{x}) = 24 \times 24$ pixels, $M_h(\mathbf{x}) = 24 \times 24$ pixels.

(vii) Repeat from (iii) to (vi) until a reasonable range of $M_h(\mathbf{x})$ is covered.

(viii) Compare the resulting $E_I[h_i]$. The pair of $M_h(\mathbf{x})$ and $M_f(\mathbf{x})$ producing the minimal $E_I[h_i]$ is chosen as the best estimates of $S_h(\mathbf{x})$ and $S_f(\mathbf{x})$ respectively.

The proposed method of estimating the supports of the image and the psf makes the iterative blind deconvolution algorithm more robust and practical. If the relative size of $S_h(\mathbf{x})$ compared with $S_f(\mathbf{x})$ is small, such as in the example shown in Figure 4.21, and also the contamination level is high (say SNR = 10dB), the error measure ceases to be a reliable guide. In these noisier cases, even the true error $E_T[f_i]$ is not a useful indicator of the correct estimate of $S_f(\mathbf{x})$ because the effect of contamination on $g(\mathbf{x})$ could be much more severe than that of the incorrect support estimate. Consequently, for contamination levels greater than approximate 10dB, the restorations, even using the exact supports, are quite poor.

Chapter 5

EXTENSIONS TO BLIND DECONVOLUTION METHODS

This chapter presents two new algorithms for iterative blind deconvolution that incorporate refinements on the Davey Algorithm discussed in §3.2.3. The motivations for these refinements are discussed in §5.1. This is followed in §5.2 by a discussion of the first new algorithm referred to as the *Automatic Iterative Algorithm* (AIA) henceforth. The AIA not only incorporates an automatic strategy to accelerate convergence, but also produces images of better quality than the Davey Algorithm. Finally, the second algorithm, referred to as the *Coloured-noise Algorithm* (CNA), is described in §5.3. This algorithm was developed to handle images contaminated by *coloured noise* (such noise is defined in §5.1).

5.1 MOTIVATION

Among the various blind deconvolution techniques reviewed in chapter 3, the Davey Algorithm (see §3.2.3) has proven to be one of the most effective. However, the need for improved techniques is highlighted by the following points.

Firstly, the Davey Algorithm does not perform optimally since it uses a fixed value for β, the filter constant (see §3.2.3). β is expected to represent two separate quantities which in general are not equal. The Fourier space constraint of the Davey Algorithm, implemented as a Wiener-like filter, is crucial to its deconvolution processing. Investigation of the role of this filter (see §4.2) reveals that: when the estimate of $F(\mathbf{u})$, $f_i(\mathbf{x})$, is to be generated, β is ideally an estimate of $|C(\mathbf{u})/F(\mathbf{u})|^{n+2}$ (see (4.58)). Similarly, when the estimate of $H(\mathbf{u})$, $h_i(\mathbf{x})$, is sought, β is related to $|C(\mathbf{u})/H(\mathbf{u})|^{n+2}$. In general, $|C(\mathbf{u})/F(\mathbf{u})|^{n+2}$ and $|C(\mathbf{u})/H(\mathbf{u})|^{n+2}$ are two different quantities. The use of a fixed value of β to simultaneously represent both of these quantities is an approximation that causes the Davey Algorithm to perform suboptimally.

Secondly, the successful performance of the Davey Algorithm may be impaired when values of the spectral estimates $|H_i(\mathbf{u})|$ and/or $|F_i(\mathbf{u})|$ are forced to be small at certain frequencies, while the true spectra $H(\mathbf{u})$ and/or $F(\mathbf{u})$ may exhibit large magnitudes at these frequencies. A Wiener-like filter plays an important role in the deconvolu-

tion processing conducted by the Davey Algorithm, as discussed in §4.2.2. This filter, $W_{\hat{H}_i}(\mathbf{u})$, acts as an inverse filter when the estimate of $|H(\mathbf{u})|$, $|\hat{H}_i(\mathbf{u})|$, is greater than the cut-off spectral magnitude H_{max} in (4.52). When $|\hat{H}_i(\mathbf{u})|$ is less than H_{max}, the value of the magnitude of $W_{\hat{H}_i}(\mathbf{u})$ is suppressed to be small which prevents exaggerating the contamination $C(\mathbf{u})$ on $G(\mathbf{u})$ in the new estimate $|F_i(\mathbf{u})|$ (see (3.20)). The value of $|F_i(\mathbf{u})|$ corresponding to this range of $|\hat{H}_i(\mathbf{u})|$ ($|\hat{H}_i(\mathbf{u})| < H_{max}$) is thus forced to be small. Although this kind of Wiener-like filter is effective for preventing contamination amplification, it may cause the loss of useful information present in $|F(\mathbf{u})|$. The same reasoning can be applied to $W_{\hat{F}_i}(\mathbf{u})$.

Thirdly, if either the psf or the image is symmetric, the algorithm performs less effectively than when both $h(\mathbf{x})$ and $f(\mathbf{x})$ are asymmetric (Bates *et al.*, 1990a).

Fourthly, energy shifting between the two estimates $\hat{h}_i(\mathbf{x})$ and $f_i(\mathbf{x})$ results during the Davey Algorithm and may cause this algorithm to perform unstably. Since the Davey Algorithm usually starts with a random image estimate $\hat{f}_0(\mathbf{x})$, many pixels within the estimates $h_i(\mathbf{x})$ and $f_i(\mathbf{x})$ violate the image space constraints. To circumvent such violation, the magnitudes of violating pixels are set to zero, which in turn sharply reduces the energy of the estimates $\hat{h}_i(\mathbf{x})$ and $\hat{f}_{i+1}(\mathbf{x})$. According to Parseval's theorem (§2.4.2.3), this also reduces the energy within their spectra $\hat{H}_i(\mathbf{u})$ and $\hat{F}_{i+1}(\mathbf{u})$, respectively, especially in the early iterations. The energy lost from one estimate, say $\hat{H}_i(\mathbf{u})$, is largely transferred to its counterpart, $F_i(\mathbf{u})$, at the corresponding frequencies, in accordance with the Fourier space constraint in (5.7). Consequently, since $\hat{h}_i(\mathbf{x})$ and $f_i(\mathbf{x})$ are related to $\hat{H}_i(\mathbf{u})$ and $F_i(\mathbf{u})$ by equation (2.17), energy shifting occurs between $\hat{h}_i(\mathbf{x})$ and $f_i(\mathbf{x})$. Similarly, energy shifting also occurs between $\hat{f}_i(\mathbf{x})$ and $h_i(\mathbf{x})$.

Finally, some form of automatic strategy needs to be devised for selecting and varying β objectively during the iterative process. In the Davey Algorithm the value of β is fixed to a constant value throughout the processing of an image and the choice of β is crucial to its successful performance. To select a suitable β for processing an image, it is customary to perform a number of trials (usually approximately 10) with different values of β. For each trial, a fixed number of iterations, usually 100, is conducted. This process is not only tedious and computationally expensive, but also relies on users to aid β selection. Also, since only a limited number of filter constants can practically be tested, the difference between selected and ideal values of β can result in significantly sub-optimal performance of the Davey Algorithm.

The above points provide the motivation for the refinements to the Davey Algorithm incorporated in the AIA discussed in the next section §5.2.

The Coloured-noise Algorithm discussed in §5.3 is motivated by the above points and the following issue. Since β in the Davey Algorithm is an estimate of $|C(\mathbf{u})/F(\mathbf{u})|^{n+2}$ or $|C(\mathbf{u})/H(\mathbf{u})|^{n+2}$ (see §4.2), it restricts the power spectrum of the contamination $|C(\mathbf{u})|^{n+2}$ to be proportional to the power spectra of both the image $|F(\mathbf{u})|^{n+2}$ and

the psf $|H(\mathbf{u})|^{n+2}$. Experiences indicate that the Davey Algorithm is also capable of blindly deconvolving images contaminated with *white noise* (i.e. the magnitude of the power spectrum of the contamination is constant at all frequencies (Castleman, 1979; Haykin, 1983)). The algorithm is therefore in theory only suitable for solving blind deconvolution problems associated with a certain type of contamination. Unfortunately, coloured noise (the power spectrum of which varies with frequency (Haykin, 1983)) is often encountered in real-world images. It is, therefore, desirable to seek techniques for coping with coloured noise contamination in order to formulate a general blind deconvolution algorithm. One such technique, and the resulting algorithm is described in §5.3.

5.2 THE AUTOMATIC ITERATIVE ALGORITHM (AIA)

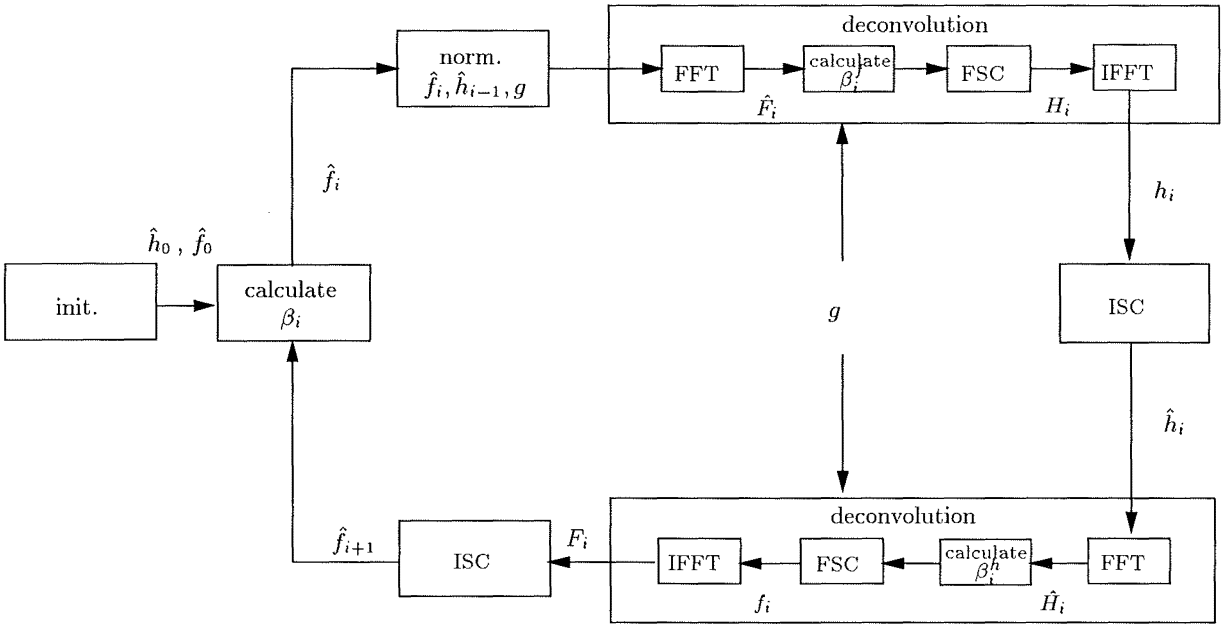
In this section, a refined algorithm is proposed which accelerates and improves the convergence of the Davey Algorithm, as well as improving its stability. This algorithm is described briefly in §5.2.1, followed by §5.2.2 and §5.2.3 which discuss the refinements it incorporates and the reasons for them. Refinements to accelerate the convergence of the algorithm are described in §5.2.2, where three alternative methods are presented along with examples to demonstrate each of the methods. Refinements to improve the quality of the restored images are described in §5.2.3. Comparative computational examples using the AIA and the Davey Algorithm are presented in §5.2.4. The results of this comparison are discussed in §5.2.5.

5.2.1 Description of the algorithm

Chapter 4 discusses the effects of key parameters on the convergence of the Davey Algorithm, including the effects of the Wiener-like filter and its constant β on the performance of this algorithm. Based on these effects and the motivations discussed in §5.1, the AIA is developed. The basic concept of this algorithm is similar to that applied by Ayers and Dainty (1988) and Davey *et al* (1989) (see Figure 3.1). Refinements are made to various aspects associated with the filter constant β, the Fourier space constraints, the normalization of estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ and the balancing of energy between $g(\mathbf{x})$ and its estimate during each iteration. Importantly, β is no longer constant in this algorithm and is henceforth referred to as the *filter parameter* and denoted β_i. The AIA is depicted in Figure 5.1; the steps in brief are:

(i) Initialization:

- Initialize the estimate of the true image $f(\mathbf{x})$, $\hat{f}_0(\mathbf{x})$, with a random image.
- Initialize the estimate of the true psf $h(\mathbf{x})$, $\hat{h}_0(\mathbf{x})$, with a random image.
- Select support estimates for $f(\mathbf{x})$ and $h(\mathbf{x})$ as described in §4.3.2.



init. = initialization
norm. = normalize
ISC = image space constraints
FSC = Fourier space constraints

Figure 5.1 The basic iterative loop of the Automatic Iterative Algorithm.

- Select initial filter parameter β_0.

(ii) Vary the filter parameter β_i. The rationale and strategies for adapting β_i are discussed in §5.2.2.

(iii) Normalize the estimates $\hat{f}_i(\mathbf{x})$ and $\hat{h}_{i-1}(\mathbf{x})$ with respect to the brightest pixels in $|\hat{f}_i(\mathbf{x})|$ and $|\hat{h}_{i-1}(\mathbf{x})|$, respectively and balance the energy of the blurred image $g(\mathbf{x})$ and the energy of its estimate $\hat{f}_i(\mathbf{x}) \odot h_{i-1}(\mathbf{x})$. See §5.2.3.2 for details.

(iv) Generate the new estimate $h_i(\mathbf{x})$ by deconvolution. This deconvolution is carried out by transforming the estimate $\hat{f}_i(\mathbf{x})$ and the blurred image $g(\mathbf{x})$ into Fourier space and then applying Fourier space constraints to produce a spectral estimate $H_i(\mathbf{u})$. $H_i(\mathbf{u})$ is then transformed back into image space to form $h_i(\mathbf{x})$. The Fourier space constraints applied are discussed further in §5.2.3.1. Note that the filter parameter involved in this Fourier space constraint is β_i^h which is calculated from β_i (*cf.* (5.7), (5.8) and (5.10)).

(v) Apply image space constraints, such as support, reality and positivity (if applicable), to the new estimate, $h_i(\mathbf{x})$, as described in §3.2.3.

(vi) Generate the next image estimates $f_i(\mathbf{x})$ and $f_{i+1}(\mathbf{x})$ from $\hat{h}_i(\mathbf{x})$ and $\hat{f}_i(\mathbf{x})$ in the same manner as the psf estimates $h_i(\mathbf{x})$ and $\hat{h}_i(\mathbf{x})$ are formed from $\hat{f}_i(\mathbf{x})$ and $\hat{h}_{i-1}(\mathbf{x})$ as described in steps (iv) and (v).

(vii) Test for termination conditions and exit if one or more are met. These conditions are:

- The error metric $E_a[g_i]$ or $E_b[g_i]$, described in (4.66) and (4.67), rises after reaching a minimum for a given number of iterations, I_D.
- The number of iterations exceeds the maximum allowable, I_{max}.

(viii) Return to (ii) and repeat loop.

Upon loop termination (see (vii)) the estimates $f_i(\mathbf{x})$ and $h_i(\mathbf{x})$ associated with the smallest error are accepted as the restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$.

Some of the steps discussed above are the same as those present in the Davey Algorithm. The following sections discuss only those steps of the AIA that incorporate refinements.

5.2.2 Refinements to accelerate convergence

The convergence of the AIA is accelerated by adapting the filter parameter β_i for the Wiener-like filter (see (5.8)). The idea of adapting β_i was first suggested by Davey (Davey, 1989, §7.4.3) and later a method for implementing it (in concept only) was proposed by Bates (Bates *et al.*, 1990a). McCallum and Rodenburg (1993) also proposed and implemented a method for adapting β_i. The McCallum and Rodenburg algorithm is reviewed in §5.2.2.1, wherein the performance of this algorithm is illustrated with examples and discussed. In §5.2.2.2 through §5.2.2.4, three alternative methods of adapting β_i in conjunction with the AIA are discussed. The first method of adapting β_i is adopted from the McCallum and Rodenburg algorithm (1993) and named the iterative β_i reduction approach. In §5.2.2.2, the application of the iterative β_i reduction approach in the AIA is investigated and a suitable reduction rate which makes the approach effective in the AIA is proposed. The second method, described in §5.2.2.3, adapts β_i using various β_i reduction rates. The convergence of the AIA is then further accelerated. The third method is to adapt β_i to minimize the error metric. This is presented in §5.2.2.4. In each of the sections §5.2.2.2 through §5.2.2.4, computational examples demonstrating the operation of the proposed methods are presented and results are discussed. The three strategies in §5.2.2.2, §5.2.2.3 and §5.2.2.4 are different approaches to solve the same problem of accelerating the convergence of the algorithm. Each of them has advantages in different contexts.

5.2.2.1 The McCallum-Rodenburg Algorithm

McCallum and Rodenburg (1993) extended the Davey Algorithm to recover probe and specimen functions from microdiffraction plane intensity measurements obtained from a scanning electron microscope. For convenience, this algorithm is referred to as the McCallum-Rodenburg Algorithm henceforth.

Under the McCallum-Rodenburg Algorithm, β_i is reduced in proportion to the iteration count i according to

$$\beta_i = \beta_0 k^i \quad 0 < k \leq 1, \tag{5.1}$$

where β_0 is the initial value of β_i and k is a real constant referred to as the *reduction rate*.

The basic structure of the McCallum-Rodenburg Algorithm is similar to the Davey Algorithm (§3.2.3). Apart from adapting β_i iteratively, two other enhancements are also made in the McCallum-Rodenburg Algorithm. Firstly, $|\hat{F}_i(\mathbf{u})|$ and $|\hat{H}_i(\mathbf{u})|$ are normalized to have a maximum value of unity before being used to generate $H_i(\mathbf{u})$ and $F_i(\mathbf{u})$, respectively. Secondly, the energies of $G(\mathbf{u})$ and $\hat{F}_i(\mathbf{u})\,\hat{H}_i(\mathbf{u})$ are normalized to unity before computing the convolutional error defined by

$$E_a'[g_i] = \int_{\mathbf{x}} |\hat{f}_i(\mathbf{x}) \odot \hat{h}_i(\mathbf{x}) - g(\mathbf{x})|^2 d\mathbf{x}\,. \tag{5.2}$$

In one of the experiments reported in McCallum and Rodenburg (1993), the McCallum-Rodenburg Algorithm was observed to perform most reliably with β_i (where $\beta_i^h = \beta_i^f = \beta_i$) decreasing with each iteration ($k = 0.8$). In another experiment reported in the same paper, the best results were obtained with constant filter parameter (i.e. $k = 1$) but $\beta_i^h \neq \beta_i^f$. McCallum and Rodenburg (1993) provided little guidance regarding whether an iterative β_i reduction scheme (i.e. equation (5.1) for $k < 1$) can be generally applied in their algorithm to produce satisfactory restorations to accelerate its convergence. As this is one of the interests in this chapter, the rest of this section is devoted to this subject.

To confirm the feasibility and reliability of employing the iterative β_i reduction approach to accelerate the performance of the McCallum-Rodenburg Algorithm, it is necessary to know whether there exist a pair of values of k and β_0 which are insensitive to the contamination level of the blurred image and from which the algorithm produces optimal or near optimal restorations, compared with those produced using other values of k. As a consequence, the following experiments were conducted.

The McCallum-Rodenburg Algorithm was applied to two blurred images contaminated with Gaussian noise to give SNRs of 20dB and 40dB, respectively. The behavior of the algorithm was investigated for seven reduction rates $k = 0.6, 0.7, 0.8, 0.9, 0.95, 0.97$ and 0.99. For each test, a blurred image $g(\mathbf{x})$ was created by convolving the true image

shown in Figure 5.2(a) (a Chinese ideogram representing luck) with the psf shown in Figure 5.2(b). An example of $g(\mathbf{x})$ contaminated with SNR of 20dB is depicted in Figure 5.2(c). As depicted in Figure 5.2, $f(\mathbf{x})$ contains 32×32 pixels and is real and positive, while $h(\mathbf{x})$ is a uniform disk with a diameter of 9 pixels.

SNR = 40dB							
Error|Iteration	k						
	0.6	0.7	0.8	0.9	0.95	0.97	0.99
$E'_{a,min}[g_i]$ †$\|I_{ming}$	4.84|7	4.66|16	4.53|25	4.4|50	4.37|98	4.35|164	4.51|400
$E_T[f_i]$ †$\|I_{ming}$	266|7	160|16	160|25	166|50	173|98	174|164	225|400
$E_{T,min}[f_i]$ †$\|I_{minf}$	160|12	158|17	158|26	159|55	161|111	162|186	225|400
SNR = 20dB							
$E'_{a,min}[g_i]$ †$\|I_{ming}$	14.83|6	14.8|8	14.8|13	14.77|29	14.75|59	14.75|99	14.75|302
$E_T[f_i]$ †$\|I_{ming}$	354|6	357|8	355|13	350|29	349.7|59	349.6|99	349|313
$E_{T,min}[f_i]$ †$\|I_{minf}$	354|6	352|9	351|14	349|30	348|62	348|104	348|313

Table 5.1 The effect of reduction rate k on minimum convolutional error $E'_{a,min}[g_i]$, the corresponding true error $E_T[f_i]$ at the same iteration I_{ming} and minimum true error $E_{T,min}[f_i]$ when using the McCallum-Rodenburg Algorithm for blind deconvolution.
† Each error value displayed is 1000 times the actual value.

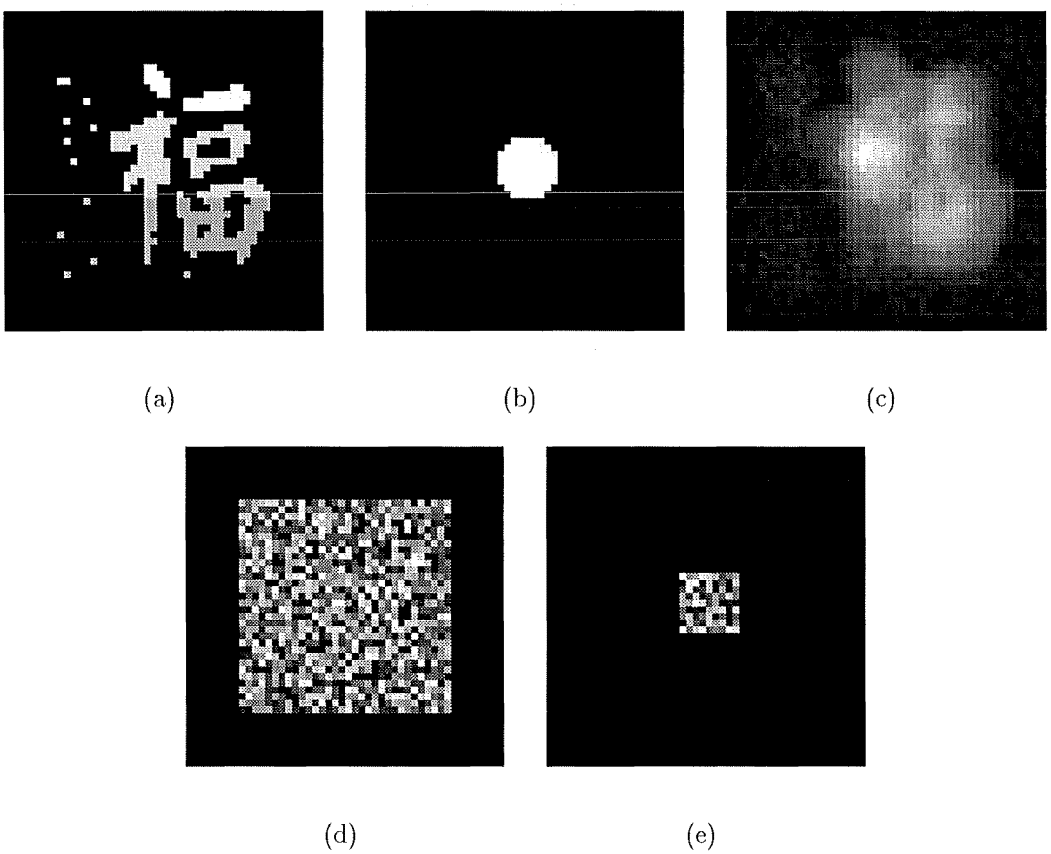
It is convenient to introduce a normalized true error, defined

$$E_T[f_i] = \frac{\int_{\mathbf{x}} |f(\mathbf{x}) - \varrho_T^i f_i(\mathbf{x})|^2 \, d\mathbf{x}}{\int_{\mathbf{x}} |f(\mathbf{x})|^2 \, d\mathbf{x}}, \tag{5.3}$$

where $\varrho_T^i = \text{Eng}[f(\mathbf{x})]/\text{Eng}[f_i(\mathbf{x})]$. Thus the energies of the two quantities in (5.3) are equal. Similarly, $E_T[h_i]$ can be calculated by substituting $f(\mathbf{x})$ and $f_i(\mathbf{x})$ for $h(\mathbf{x})$ and $h_i(\mathbf{x})$ respectively in (5.3).

When performing the McCallum-Rodenburg Algorithm, the initial estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$, shown in Figures 5.2(d) and 5.2(e), were generated by pseudo-random numbers uniformly distributed between 0 and 1 within the support estimates $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$, respectively. $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$ were selected to equal the image box (*cf.* §2.2). After initial trials a value of $\beta_0 = 0.1$ was selected and found to be satisfactory for all images and values of k tested. The maximum iteration was set to 500 (the optimal restorations occur within this iteration count). The restorations were obtained at iteration I_{ming} where the corresponding convolutional error $E'_a[g_i]$ was minimum, that is $E'_a[g_i] = E'_{a,min}[g_i]$. Table 5.1 lists the observed values of the error $E'_{a,min}[g_i]$ and the corresponding true error $E_T[f_i]$ at iteration I_{ming} for each contamination level and reduction rate considered. Also listed is the minimum true error, $E_{T,min}[f_i]$, observed over all 500 iterations and the iteration I_{minf} at which it occurs. Importantly, I_{minf}, can and

often does differ from I_{ming}, since $E_a'[g_i]$ evaluates the difference between the contaminated measurement $g(\mathbf{x})$ and its estimate, while $E_T[f_i]$ assesses the difference of the true image $f(\mathbf{x})$ and its estimate. Figure 5.3 shows examples of the restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ obtained using the McCallum-Rodenburg Algorithm. These restorations are the best observed in conjunction with the various reduction rates tested. Figures 5.3(a) and 5.3(b) resulted when $k = 0.8$ and $g(\mathbf{x})$ has a SNR of 40dB. Figures 5.3(c) and 5.3(d) resulted when $k = 0.95$ and $g(\mathbf{x})$ has a SNR of 20dB. The error curves of $E_a'[g_i]$ and $E_T[f_i]$ associated with Figures 5.3(a) and 5.3(b) are plotted in Figure 5.4(a), and those corresponding to Figures 5.3(c) and 5.3(d) are plotted in Figure 5.4(b).

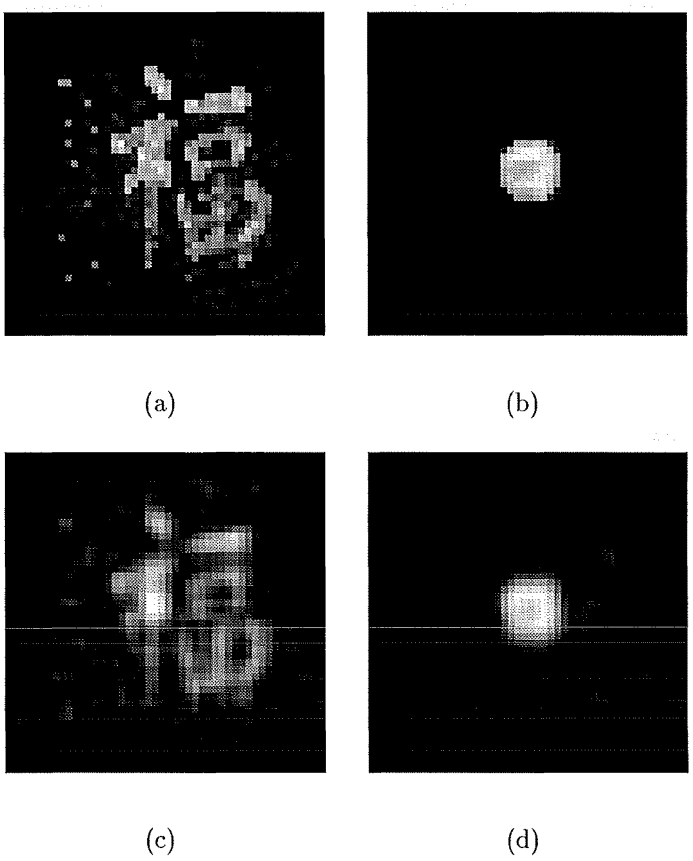


(a) (b) (c)

(d) (e)

Figure 5.2 (a) A real positive image $f(\mathbf{x})$; (b) psf $h(\mathbf{x})$; (c) blurred image $g(\mathbf{x}) = f(\mathbf{x}) \odot h(\mathbf{x}) + c(\mathbf{x})$ with SNR = 20dB; (d) and (e) starting image $\hat{f}_0(\mathbf{x})$ and psf $\hat{h}_0(\mathbf{x})$.

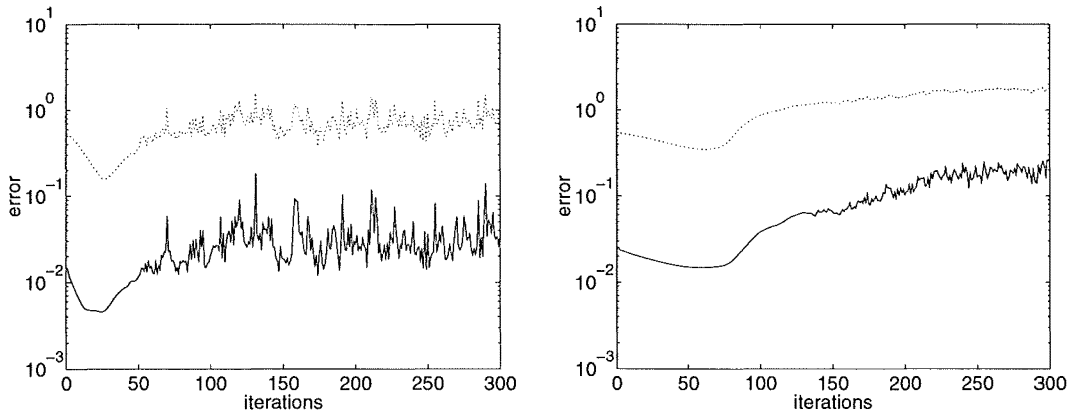
Inspection of the results depicted in Figure 5.3 and Table 5.1 reveals that when the initial value of β_i, β_0, is appropriate (for example $\beta_0 = 0.1$), the errors $E_{a,min}'[g_i]$, $E_T[f_i]$ and $E_{T,min}[f_i]$ observed for $g(\mathbf{x})$ with both SNRs of 20dB and 40dB are insensitive to the value of k selected, when it is in the range of $0.7 - 0.97$. Using this range of k, the algorithm produces similar and low values of the errors of the restorations, and the observed minimum error $E_{a,min}'[g_i]$ correlates well with the minimum true error $E_{T,min}[f_i]$, which makes it practical to obtain the best or near the best restorations produced by the algorithm. Such response of the algorithm to the concerned range of



(a) (b)

(c) (d)

Figure 5.3 Examples of images restored using the McCallum-Rodenburg algorithm, $\beta_0 = 0.1$: restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ corresponding to Figure 5.2 respectively with $g(\mathbf{x})$ contaminated by: (a) and (b) $k = 0.8$, SNR = 40dB; (c) and (d) $k = 0.95$, SNR = 20dB.



(a) (b)

Figure 5.4 The convolutional error $E_a'[g_i]$ and the true error $E_T[f_i]$ corresponding to the restorations depicted in Figure 5.3: (a) $k = 0.8$, SNR = 40dB; and (b) $k = 0.95$, SNR = 20dB. Continuous and dotted curves depict $E_a'[g_i]$ and $E_T[f_i]$ respectively.

k is consistent for $g(\mathbf{x})$ contaminated with both levels. It is reasonable and practical to infer that this behaviour of the algorithm is in accord for $g(\mathbf{x})$ contaminated with various levels. The McCallum-Rodenburg Algorithm is, therefore, capable of producing satisfactory restorations as a consequence of using iterative β_i reduction approach and performing one application, for example using $k = 0.9$ and $\beta_0 = 0.1$. In contrast, the Davey Algorithm must typically be applied ten times with different fixed constant β to produce acceptable restorations.

5.2.2.2 Applying iterative β_i reduction in the Automatic Iterative Algorithm: method one

The results presented in the previous section show that the convergence of the McCallum-Rodenburg Algorithm is accelerated by reducing β_i iteratively. However, the literature concerning blind deconvolution provides little information regarding the rationale for reducing β_i; this is now discussed.

As discussed in §4.2.2, β is an estimate of $|C(\mathbf{u})|^2/|F(\mathbf{u})|^2$ (see (4.58) where $n = 0$) when the estimate $F_i(\mathbf{u})$ is generated using the Davey Algorithm. The value for β selected determines the value of the cutoff spectral magnitude H_{max} (*cf.* (4.52)), which separates the effective part of the spectrum $\hat{H}_i(\mathbf{u})$ from its ineffective part, at the commencement of image restoration. Since most of the effective components are very noisy in the early iterations and are retrieved gradually during the iterative process, the value of the actual cutoff spectral magnitude decreases from a large value to the value of H_{max}. If β is a suitable choice, the estimate of $F(\mathbf{u})$ is then retrieved. A similar process occurs when the estimate of $H(\mathbf{u})$ is retrieved.

The iterative blind deconvolution process stated in the previous paragraph suggests that reducing β_i iteratively as described in (5.1) is suitable for the deconvolution process. Starting with a larger value of β_0 which is appropriate to the noisy starting estimate, the iterative β_i reduction process forces the amount of the actual effective components of the estimate, say $\hat{H}_i(\mathbf{u})$, to be increased at a rate that is compatible with the β_i reduction rate and retrieves these components progressively. The true error of the image $E_T[f_i]$ decreases accordingly. This process continues until β_i is reduced to a value that results in the lowest value of $E_T[f_i]$. This value of β_i is called the optimum value of β_i. It is likely that the optimum value of β_i produces a pair of constant β_i^f and β_i^h (see §5.2.3.3) which are most close to their representative pair, the averages of $|C(\mathbf{u})|^2/|H(\mathbf{u})|^2$ and $|C(\mathbf{u})|^2/|F(\mathbf{u})|^2$. When β_i is further reduced to be smaller than its optimum, the contamination in $G(\mathbf{u})$ is likely to be overemphasised in the new estimate $F_i(\mathbf{u})$. The error curve of $E_T[f_i]$ rises correspondingly as shown in Figure 5.4 and the algorithm is said to diverge. The direct connection between β_i, the residual $C_i(\mathbf{u})$ and the cutoff spectral magnitude at each iteration, H_{imax}, is described mathematically in §5.3.1 (see (5.16) and (5.18)). The iterative β_i reduction approach of the McCallum-Rodenburg

Algorithm presented in one of their examples (McCallum and Rodenburg, 1993) is, therefore, appropriate to be employed.

SNR = 40dB							
Error|Iteration	k						
	0.6	0.7	0.8	0.9	0.95	0.97	0.99
$E'_{a,min}[g_i]$ †$\|I_{ming}$	4.17|9	3.59|12	3.0|21	1.34|52	1.01|101	0.771|196	23.4|3
$E_T[f_i]$ †$\|I_{ming}$	386|9	368|12	291|21	81.4|52	71.6|101	32.9|196	792|3
$E_{T,min}[f_i]$ †$\|I_{minf}$	366|11	312|17	124|36	54.9|62	45.9|118	32.3|204	585|26
SNR = 20dB							
$E'_{a,min}[g_i]$ †$\|I_{ming}$	11.3|8	10.9|12	10.7|18	10.4|43	9.67|85	9.50|140	33.2|3
$E_T[f_i]$ †$\|I_{ming}$	519|8	502|12	452|18	368|43	286|85	258|140	794|3
$E_{T,min}[f_i]$ †$\|I_{minf}$	517|7	486|11	449|17	366|44	282|83	248|134	589.9|26

Table 5.2 The effect of reduction rate k on the errors $E'_{a,min}[g_i]$, $E_T[f_i]$ and $E_{T,min}[f_i]$ produced by the Automatic Iterative Algorithm for blind deconvolution.
† Each error value displayed is 1000 times the actual value.

Based on the above discussion, the approach of iterative β_i reduction is employed in the AIA described in §5.2.1. Using the same example shown in Figures 5.2(a) to 5.2(d), the algorithm was applied to the blurred image in Figure 5.2(c) in one case contaminated by SNR = 40dB and in a second by 20dB. The reduction rates $k = 0.6, 0.7, 0.8, 0.9, 0.95, 0.97$ and 0.99 were each used. The minimum of the error $E_b[g_i]$ over 300 iterations in each case is shown in Table 5.2. The corresponding true error of the image $E_T[f_i]$ and the minimum $E_T[f_i]$ are shown as well. Restorations with $k = 0.97$, which are the best among the tested values of k, are shown in Figure 5.5 and the corresponding error curves of $E_b[g_i]$ and $E_T[f_i]$ are plotted in Figure 5.6.

Inspection of the error values obtained in different cases in Table 5.2 reveals three things: Firstly, unlike the results using the McCallum-Rodenburg Algorithm in Table 5.1, the values of $E_b[g_i]$ and $E_T[f_i]$ in Table 5.2 are largely dependent on the reduction rate k. Generally speaking, the slower the reduction rate is, i.e. the larger value of k, the lower values of $E_b[g_i]$ and $E_T[f_i]$ can be obtained. Equivalently, the better is the convergence of the algorithm. However, if the reduction rate is too slow, for example, $k = 0.99$, the algorithm ceases to converge and the deconvolution therefore fails.

Secondly, the required iteration number to achieve acceptable restorations depends on the contamination level of $g(\mathbf{x})$. The less $g(\mathbf{x})$ is contaminated, the greater the number of iterations required. For example, when $k = 0.97$ the number of iterations to achieve minimum $E_T[f_i]$ over 300 iterations is 134 when SNR = 20dB and is 204 when SNR = 40dB.

Thirdly, Figure 5.6 illustrates that $E_b[g_i]$ correlates well with $E_T[f_i]$, but the iteration number at which each reaches a minimum varies with the reduction rate k. The

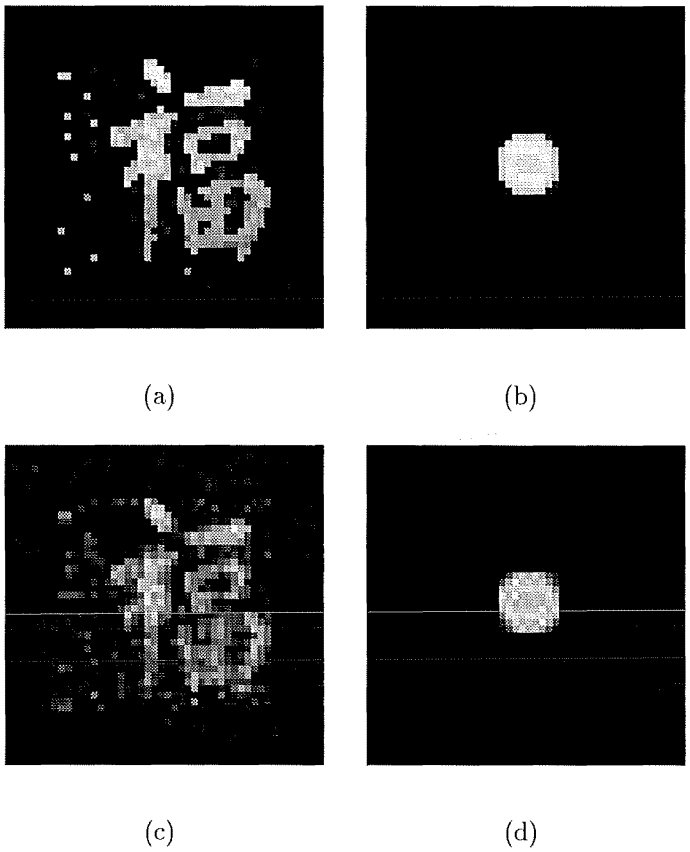


Figure 5.5 Examples of accelerating convergence by the Automatic Iterative Algorithm, using iterative β_i reduction scheme, $\beta_0 = 0.1$, $k = 0.97$: restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ corresponding to Figure 5.2 respectively with $g(\mathbf{x})$ contaminated by: (a) and (b) SNR = 40dB; (c) and (d) SNR = 20dB.

difference in one case shown in Table 5.2 is 8 iterations ($k = 0.97$, SNR = 40dB) and in another 17 iterations ($k = 0.95$, SNR = 40dB). The resultant restorations obtained from the minimum $E_b[g_i]$ thus deviate from the optima resulting from minimum $E_T[f_i]$. However, the degree of the deviation in the quality of the restorations tends to reduce as the reduction rate k gets closer to unity.

The reason for the behavior of the algorithm using different reduction rate k is now explained. As it has been discussed in the beginning of this section that reducing β_i iteratively sets a decreasing cutoff spectral magnitude H_{imax} in the estimate $\hat{H}_i(\mathbf{u})$. More and more components of $\hat{H}_i(\mathbf{u})$ are retrieved and classified into the category of effective components (for which $|\hat{H}_i(\mathbf{u})| > H_{imax}$) as β_i decreases. Such a process ceases once β_i is reduced beyond its optimum value, which is related to $|C(\mathbf{u})|^2/|F(\mathbf{u})|^2$. If the contamination level is lower, the average of $|C(\mathbf{u})|$ is smaller, so is $|C(\mathbf{u})|/|F(\mathbf{u})|$. The optimum value that β_i is reduced to is smaller correspondingly. For the same value of the reduction rate k, more iterations are thus required.

When a slower β_i reduction rate (or a larger value of k) is selected, the difference

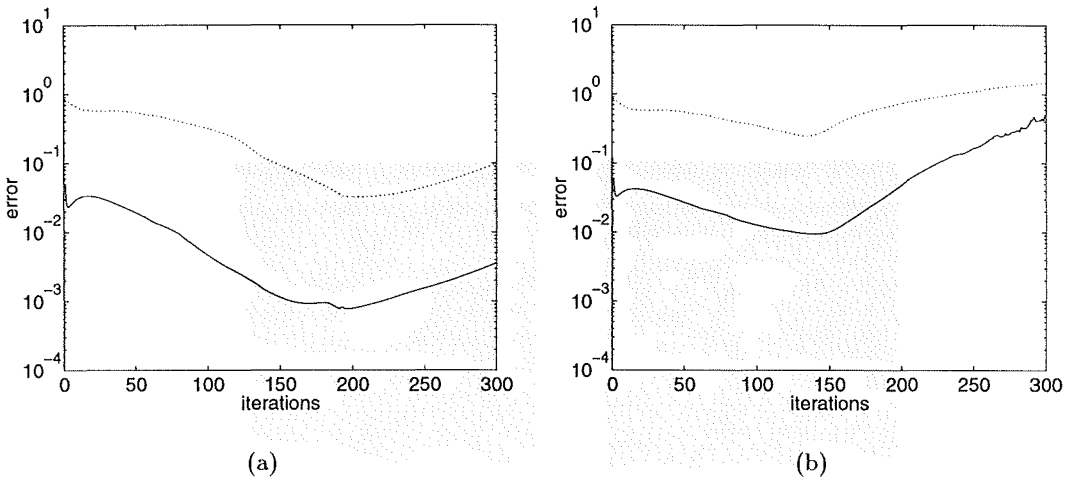


Figure 5.6 Error curves of $E_b[g_i]$ and $E_T[f_i]$ using the automatic iterative algorithm, $\beta_0 = 0.1$, $k = 0.97$, corresponding to the restorations in Figure 5.5 with (a) SNR = 40dB; and (b) SNR = 20dB. Continuous and dotted curves depict $E_b[g_i]$ and $E_T[f_i]$ respectively.

between two filter constants for two consecutive iterations β_{i-1} and β_i is smaller. The components of the interpolated new estimate, say $F_i(\mathbf{u})$, which are either generated by the Wiener-like filter using β_i or taken from those of the previous estimate $\hat{F}_i(\mathbf{u})$ related to β_{i-1}, are more smoothly merged (see §5.2.3.1). The resultant restorations are thus better retrieved than those from a faster reduction rate.

Comparison of the error values in Tables 5.1 and 5.2 and the restorations in Figures 5.5 and 5.3 shows that better restorations or equivalently lower values of $E_T[f_i]$ are from Table 5.2 produced by the AIA. The advantage of the AIA over the McCallum-Rodenburg Algorithm is shown by the significant improvement of the convergence of the algorithm especially when the contamination level is around SNR = 40dB or better.

5.2.2.3 Adapting the rate of β_i reduction: method two

Section §5.2.2.2 has demonstrated that the convergence of the Automatic Iterative Algorithm can be significantly accelerated using the iterative β_i reduction approach. Since the contamination level of the given blurred image is unknown *a priori*, a large number of iterations is required when using the recommended slow reduction rate $k = 0.97$ to operate the process as shown in Table 5.2. It is therefore worthwhile to seek the alternative especially when the size of the image is comparatively large.

In this section, a strategy to further accelerate the convergence of the algorithm based on the β_i reduction approach is presented. Since the initial filter constant β_0 is much larger than its optimum, it is sensible to initially reduce β_i rapidly and slow the rate of decrease when β_i is getting closer to its optimum. Less iterations are required to

decrease β_i from β_0 to its optimum. The convergence of the algorithm is thus further accelerated, while the advantages of using slow reduction rate discussed in §5.2.2.2 are retained.

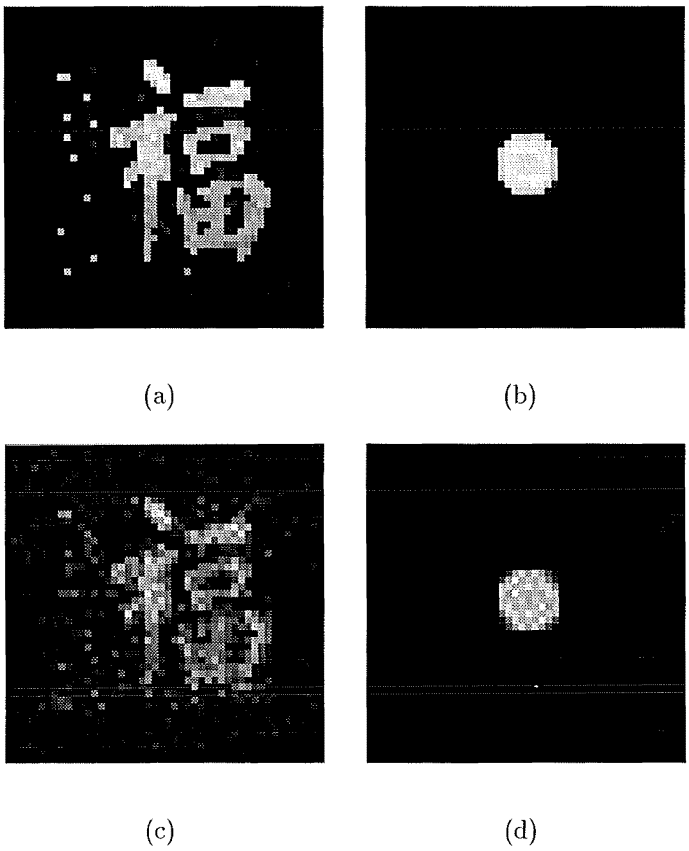


Figure 5.7 Examples of accelerating convergence by the Automatic Iterative Algorithm, using three β_i reduction rates $k = 0.9$, $k = 0.95$ for 30 iterations and $k = 0.97$ for 50 iterations consecutively: restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ corresponding to Figure 5.2 respectively with $g(\mathbf{x})$ contaminated by: (a) and (b) SNR = 40dB; (c) and (d) SNR = 20dB.

Computational simulation was carried out by adapting several β_i reduction rates consecutively to perform the algorithm. For the convenience of comparison, the same example involved in Table 5.2 was used. The true image, psf, the blurred image and the pseudo-random starting image were those displayed in Figures 5.2(a) to (d). Three different reduction rates k were chosen; the algorithm starts with $k = 0.9$ for 30 iterations, continues with $k = 0.95$ for 30 iterations and with $k = 0.97$ for a final 50 iterations. The restorations corresponding to the minimum $E_b[g_i]$ over the total 110 iterations were obtained. The same process was applied for deconvolving images contaminated by SNR = 40dB and 20dB. Restorations retrieved in the two cases are shown in Figures 5.7(a) and 5.7(b). The related error curves of $E_b[g_i]$ and $E_T[f_i]$ are displayed in Figures 5.8(a) and 5.8(b). The true error $E_T[f_i]$ when using sole rate of $k = 0.97$ in Section 5.2.2.2 is shown for comparison.

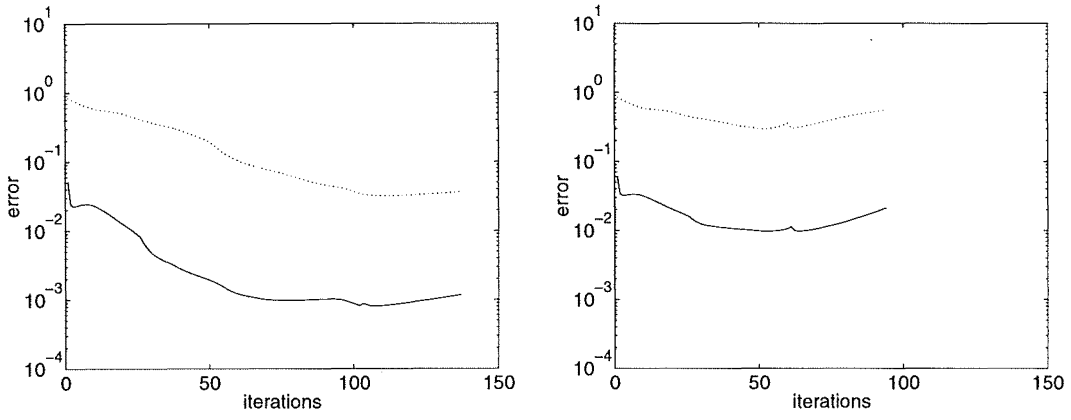


Figure 5.8 Error curves of $E_b[g_i]$ and $E_T[f_i]$ using the automatic iterative algorithm with the scheme of adapting β_i reduction rate, corresponding to the restorations in Figure 5.7 with (a) SNR = 40dB; and (b) SNR = 20dB. Continuous and dotted curves depict $E_b[g_i]$ and $E_T[f_i]$ respectively.

Observation of Figures 5.8(a) and 5.8(b) shows that by reducing β_i with $k = 0.9$, the initial rate of decrease in error $E_T[f_i]$ can be improved. In other words, the convergence of the algorithm is accelerated. When k is altered from 0.90 to 0.95, the rate of decrease in $E_T[f_i]$ slows as well. However, it is still decreasing faster than when k is held at 0.97 throughout the process. As β_i is reduced near to its optimum, or equivalently, as $E_b[g_i]$ approaches its minimal value, k is further altered to the reliable rate of 0.97. Overall the minimum achieved with the stepwise increases in k is the same as when k was fixed at 0.97, yet it was achieved 47% faster in the 40dB case, 39% faster in the 20dB case.

Since the iterations required to obtain satisfactory restorations vary with the contamination level, it is sensible to set some condition to terminate the process. Investigation of the behavior of the algorithm shows that, generally speaking, the error metric $E_b[g_i]$ decreases monotonically until reaching its minimum. It is therefore reasonable to terminate the algorithm where $E_b[g_i]$ rises above its minimum after a certain number of iterations I_D, for example $I_D = 50$. Unnecessary iterations can therefore be avoided. Both the reliability and the speed of the performance of the algorithm are improved simultaneously.

5.2.2.4 Adaptation of β_i to minimize the error metric: method three

In this section a third alternative approach is presented for dynamically changing the filter constant and accelerating the convergence of the algorithm. Instead of reducing β_i iteratively as described in §5.2.2.2 and §5.2.2.3, the new approach varies β_i according to the contamination level of $G(\mathbf{u})$ as well as the quality of the estimate $\hat{H}_i(\mathbf{u})$ during

the process. This is realised by selecting β_i to minimize the error metric $E_b[g_i]$ during various stages of the process. The algorithm is therefore directed to the solution of the deconvolution process routinely and automatically.

For convenience, the basic loop of the Automatic Iterative Algorithm utilizing the scheme of adapting β_i to minimize the error metric $E_b[g_i]$, also referred as the minimization method, is illustrated in Figure 5.9.

The difference between the two iterative loops in Figure 5.1 and Figure 5.9 is that the latter involves more steps for the scheme of dynamically changing β_i than the former does. Since the remainder of the algorithm has been described in §5.2.1, only the additional steps for adapting β_i to minimize the error metric $E_b[g_i]$, corresponding to the step (ii) in §5.2.1, are described:

(1) At the point in the iteration loop where a new image estimate $f_i(\mathbf{x})$ has been formed (*cf.* . Figure 5.9), search for a new β_i if either of the following conditions are met:

- The number of iterations performed by the same filter parameter is equal to a given number I_s;
- The algorithm has diverged beyond the tolerance, that is the algorithm produces monotonically increasing $E(g_i) > E(g_{i-1})$ for I_{div} successive iterations.

(2) Calculate a testing filter parameter $\beta_{i,j}$ by

$$\beta_{i,j} = \beta_i 10^{j/2^{l-1}}, \tag{5.4}$$

where j is an integer and is given by

$$j = 0,\ \pm 1,\ \pm 2\ ,..,\ \pm(J-1)/2,\ J \text{ odd}, \tag{5.5}$$

where J is the total number of $\beta_{i,j}$ tested in the search. The integer l in (5.4) is used to control the range of values for $\beta_{i,j}$ that are tested and is given by

$$l = \begin{cases} \textit{integer part of } (i/I_s), & i \geq I_s \\ 1 & \text{otherwise.} \end{cases} \tag{5.6}$$

where I_s is defined in step (1). l increases as iterations proceed as shown in (5.6). Consequently, the gap between two adjacent testing parameters $\beta_{i,j}$ and $\beta_{i,j+1}$ in (5.4) becomes smaller or finer as l increases.

(3) Employ the testing parameter $\beta_{i,j}$ to calculate $\beta_{i,j}^h$ according to (5.10) which is then used in the Fourier space constraint in (5.8) to generate the testing estimate $\hat{F}_{i,j}(\mathbf{u})$ and then $\hat{f}_{i,j}(\mathbf{x})$.

(4) Preserve the minimum error $E(g_{i,j_o})$ among $E(g_{i,j})$ resulted from all of the tested $\beta_{i,j}$, and the corresponding β_{i,j_o}.

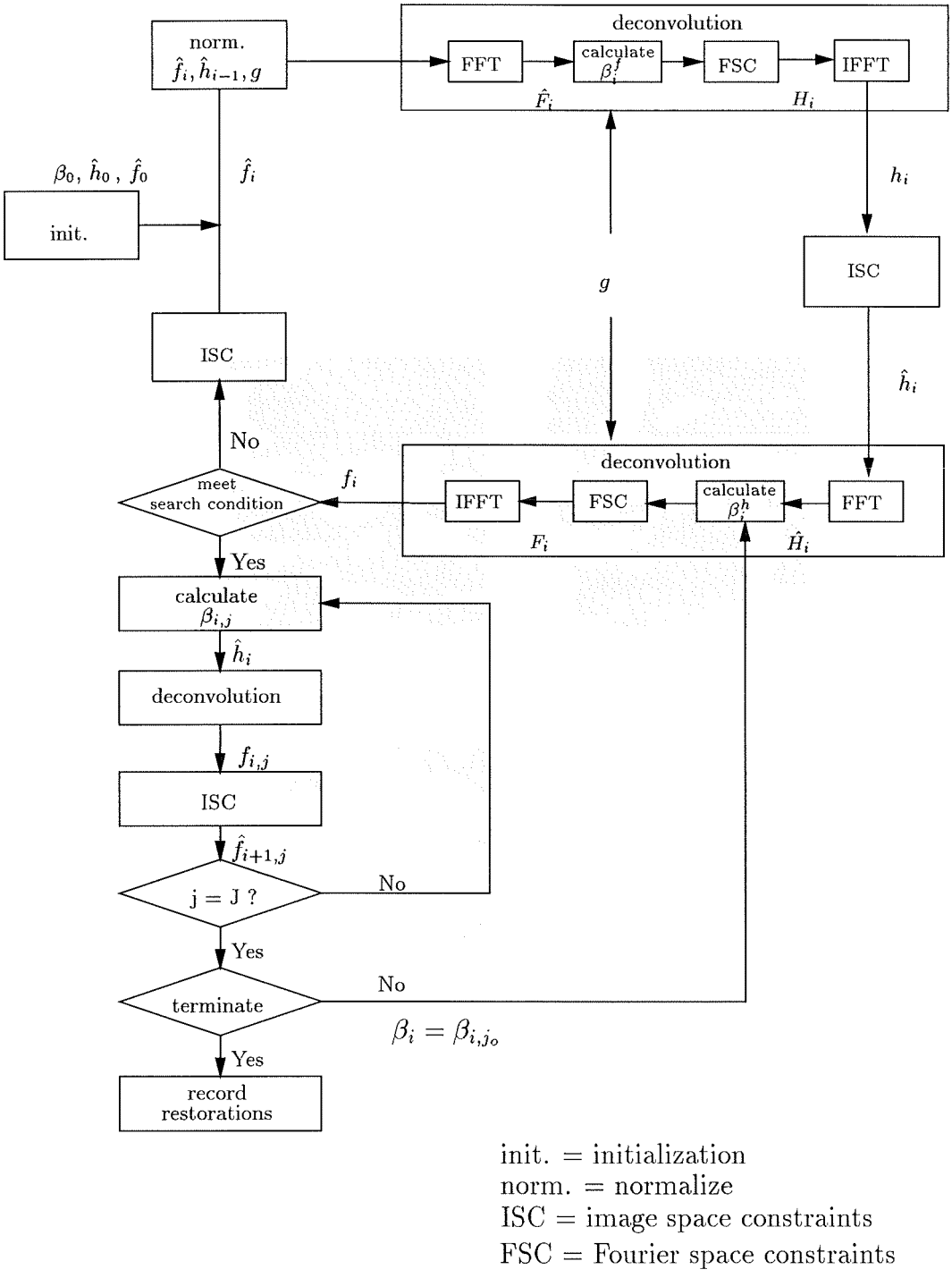


Figure 5.9 The basic iterative loop of the Automatic Iterative Algorithm, incorporating with the scheme of adapting β_i to minimize the error metric.

(5) If $j < (J-1)/2$, return to step (2), otherwise iterations cease.

(6) If $E(g_{i,jo}) < E(g_{i-1})$, go to step (7), otherwise, all of the testing $\beta_{i,j}$ result in increased values of the error metric, that is $E(g_{i,jo}) > E(g_{i-1})$; the algorithm is interpreted to have reached its convergence limit and iterations cease.

(7) Replace the previous executive β_i, which lead to either a slower convergence rate or the divergence of the algorithm at i^{th} iteration, by the optimal filter parameter $\beta_{i,jo}$ at i^{th} iteration, and return to the main loop to generate the new estimate $F_i(\mathbf{u})$, etc .

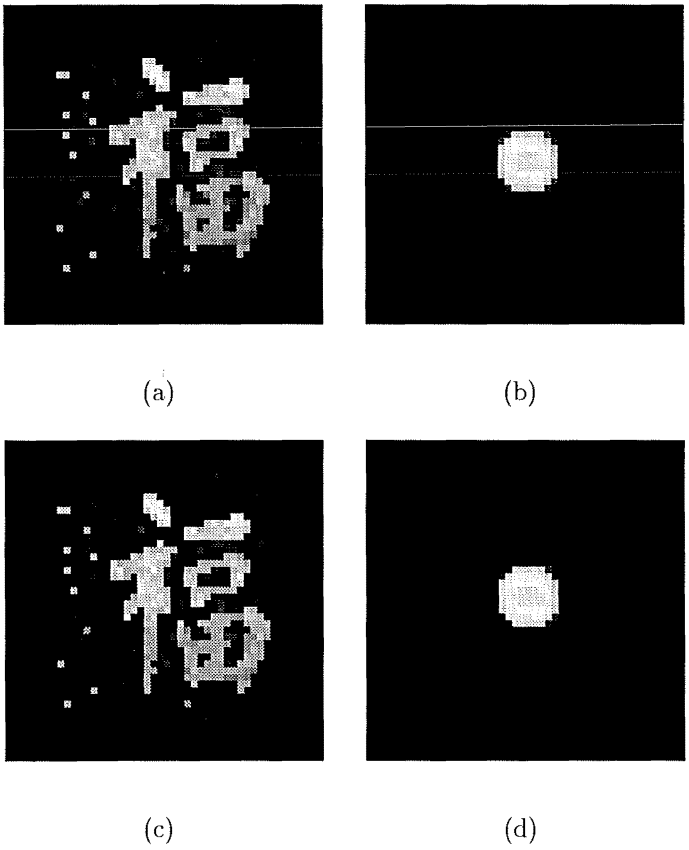
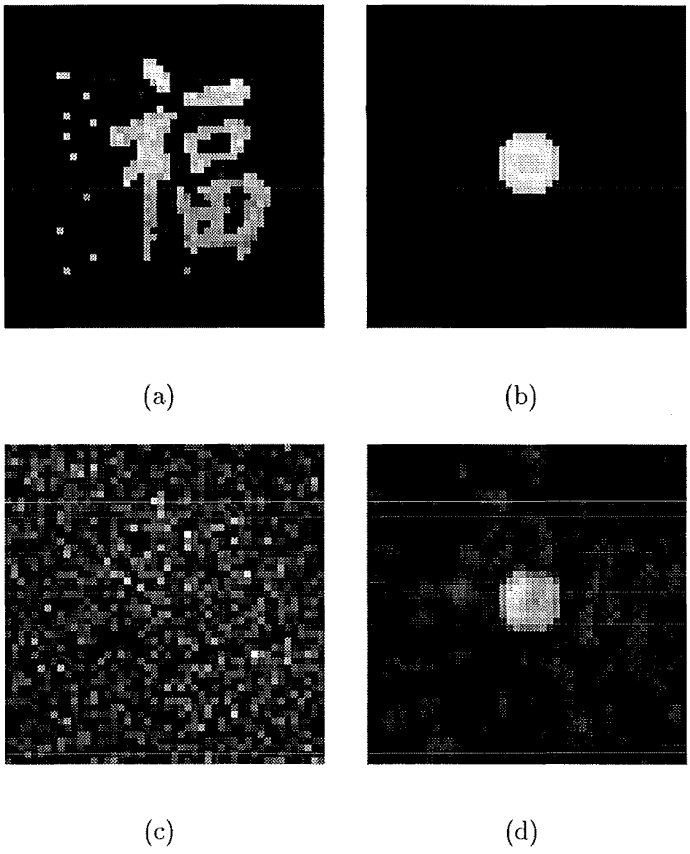


Figure 5.10 Comparative example of accelerating convergence by the automatic algorithm using the scheme of adapting β_i to minimize the error metric, SNR = 40dB, restorations of Figures 5.2(a) and 5.2(b), respectively, with initial filter parameter: (a) and (b) $\beta_0 = 10^{-1}$; (c) and (d) $\beta_0 = 10^{-7}$.

An example of the application of the AIA employing the strategy of adapting β_i to minimize $E_b[g_i]$ is now presented. Results are compared with that from the algorithm using the iterative β_i reduction approach in §5.2.2.2. Using the example displayed in Figures 5.2(a) to 5.2(d), the blurred image $g(\mathbf{x})$ was contaminated to a level of SNR = 40dB. The algorithm was run with $I = 7, J = 7$ and $I_{div} = 3$. The initial filter parameter is selected as β_0 is selected to be 10^{-1} and 10^{-7} respectively. Restorations resulted from

the minimization method are illustrated in Figure 5.10. Restorations using iterative β_i reduction scheme described in §5.2.2.2 are shown in Figures 5.11(a) to 5.11(d).



(a) (b)

(c) (d)

Figure 5.11 Comparative example of accelerating convergence by the automatic iterative algorithm employing the iterative β_i reduction approach, $k = 0.97$, SNR = 40dB: restorations of Figures 5.2(a) and 5.2(b), separately, with: (a) and (b) $\beta_0 = 0.1$; (c) and (d) $\beta_0 = 10^{-7}$.

Inspection of the results presented in Figure 5.10 reveals that successful blind deconvolution can be achieved using the initial filter parameters covering a wide range of $\beta_0 = 10^{-7}$ and 10^{-1}, i.e. the algorithm is relatively insensitive to the choice of β_0. When the same range of β_0 was used by the iterative β_i reduction scheme in §5.2.2.2, restorations (displayed in Figures 5.11(a) and (b)) obtained using the larger value of $\beta_0 = 10^{-1}$ are comparable with those using the minimization method. However, the restorations received from the smaller value of $\beta_0 = 10^{-7}$ are highly contaminated as shown in Figures 5.11(c) and 5.11(d) and are much worse than those from the minimization method with the same value of β_0.

The comparative results show that using the scheme of selecting β_i to minimize the error metric, the convergence of the algorithm, judged by the lowest value of the true error $E_T[f_i]$ that the algorithm is able to achieve, is similar to that from the iterative β_i reduction method. The advantage of the presented method is that it makes the algorithm far less sensitive to the choice of initial filter constant.

Experiences show that the algorithm can produce satisfactory restorations to images contaminated with SNR = 30dB or better. It is expected that a more severely contaminated image, e.g. SNR = 20dB, might be successfully retrieved if the algorithm is further refined.

5.2.3 Refinements for improving the accuracy of restoration

In this section several extensions to the Davey Algorithm are presented for improving the quality of the restorations. An interpolated Fourier space constraint is introduced in §5.2.3.1. A new scheme of normalizing the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$, and balancing the energy of the measurement $g(\mathbf{x})$ and its estimate is described in §5.2.3.2. In §5.2.3.3 a method is proposed for scaling the filter parameter β_i to adapt to different magnitude ranges of the estimates $|\hat{F}_i(\mathbf{u})|$ and $|\hat{H}_i(\mathbf{u})|$ when β_i is used to generate $H_i(\mathbf{u})$ and $F_i(\mathbf{u})$ respectively. In each of the sections, the significance of the refinements is given. Comparative simulation results for demonstrating their effects are presented in §5.2.4.

5.2.3.1 An interpolated Fourier space constraint

It has been discussed in §3.1.1 that at the frequencies where the spectrum of the blurred image, $G(\mathbf{u})$, is dominated by the contamination (*cf.* (3.5)), $G(\mathbf{u})$ is unable to supply useful information of $F(\mathbf{u})\,H(\mathbf{u})$. Consequently, information of $F(\mathbf{u})$ and $H(\mathbf{u})$ at these frequency region is irretrievable from $G(\mathbf{u})$. It is also discussed in §5.1 that the application of the Wiener-like filter in the Davey Algorithm makes good use of the valuable information of $G(\mathbf{u})$ to retrieve estimates of $F(\mathbf{u})$ and $H(\mathbf{u})$ respectively and prevents contamination amplification on the estimates of $F(\mathbf{u})$ and $H(\mathbf{u})$ at the cost of discarding the spectral information of the estimates at the related frequencies.

Since the iterative algorithm enforces the image and Fourier space constraints alternatively to the new estimates $H_i(\mathbf{u})$, $h_i(\mathbf{x})$, $F_i(\mathbf{u})$ and $f_i(\mathbf{x})$, respectively, at each iteration, both constraints often deal with different set of data of, for example, $H_i(\mathbf{u})$ and $h_i(\mathbf{x})$. At the frequencies where $G(\mathbf{u})$ and the Wiener-like filter contribute little to the new estimates $H_i(\mathbf{u})$ and $F_i(\mathbf{u})$, their previous estimates $\hat{H}_{i-1}(\mathbf{u})$ and $\hat{F}_i(\mathbf{u})$ may provide valuable information, because the inverse Fourier transforms of $\hat{H}_{i-1}(\mathbf{u})$ and $\hat{F}_i(\mathbf{u})$ satisfy the image space constraints. Information of $F(\mathbf{u})$ and $H(\mathbf{u})$ beyond a bandlimit of $G(\mathbf{u})$, where $G(\mathbf{u})$ is under the control of contamination, may therefore be recovered. A form of interpolation between the previous and new estimates is consequently suggested. Something of this kind does occur in the Ayers and Dainty algorithm (*cf.* §3.2.2) as shown in (3.14). Inspired by McCallum's suggestion (McCallum 1990, §8.2.2), which does not appear to have been implemented effectively yet, a similar form of interpolation of $F_i(\mathbf{u})$ is realised by

$$F_i(\mathbf{u}) = \lambda_{\hat{H}_i}(\mathbf{u})W_{\hat{H}_i}(\mathbf{u})G(\mathbf{u}) + (1 - \lambda_{\hat{H}_i}(\mathbf{u}))\hat{F}_i(\mathbf{u}), \quad 0 < \lambda_{\hat{H}_i}(\mathbf{u}) < 1, \tag{5.7}$$

where $W_{\hat{H}_i}(\mathbf{u})$ takes the form of the Wiener-like filter in (3.19) with the exponent n set to zero,

$$W_{\hat{H}_i}(\mathbf{u}) = \frac{\hat{H}_i^*(\mathbf{u})}{|\hat{H}_i(\mathbf{u})|^2 + \beta_i^h}, \tag{5.8}$$

where β_i^h is the filter parameter for producing the estimate of $F(\mathbf{u})$. β_i^h is calculated from β_i as described in §5.2.3.3.

In (5.7) $\lambda_{\hat{H}_i}(\mathbf{u})$ is a weight function, similar in form to a highpass Butterworth filter of order m (Gonzalez and Woods, 1992),

$$\lambda_{\hat{H}_i}(\mathbf{u}) = \frac{1}{1 + \mu[\beta_i^h / |\hat{H}_i(\mathbf{u})|^2]^m}, \tag{5.9}$$

$\lambda_{\hat{H}_i}(\mathbf{u})$ varies from one to zero according to the usefulness of each of the two items in (5.7). More details regarding the effect of $\lambda_{\hat{H}_i}(\mathbf{u})$ in (5.7) are described in the following paragraph. In (5.9) μ is a real constant with $0 < \mu \leq 1$ and m is an integer. μ affects the size of $\lambda_{\hat{H}_i}(\mathbf{u})$. The exponent m is responsible to the transmission slope of $\lambda_{\hat{H}_i}(\mathbf{u})$ varying from zero to one. The higher the order m, the sharper the slope. For simplicity, both μ and m are set to unity.

The rationale of the interpolated Fourier space constraint in (5.7) is now explained. At the frequencies where $|\hat{H}_i(\mathbf{u})|^2 \gg \beta_i^h$, i.e. $\hat{H}_i(\mathbf{u})$ is effective, $W_{\hat{H}_i}(\mathbf{u})G(\mathbf{u})$ in the first item of (5.7) supplies valuable information to the estimate $F_i(\mathbf{u})$. In this situation, the weight $\lambda_{\hat{H}_i}(\mathbf{u})$ is close to unity which allows the useful information from $W_{\hat{H}_i}(\mathbf{u})$ and $G(\mathbf{u})$ to be fully used in generating $F_i(\mathbf{u})$. When $|\hat{H}_i(\mathbf{u})|^2 \ll \beta_i^h$, i.e. $\hat{H}_i(\mathbf{u})$ is non-effective, the magnitude of $W_{\hat{H}_i}(\mathbf{u})\,G(\mathbf{u})$ is attenuated to prevent contamination amplification in $F_i(\mathbf{u})$ rather than provides useful information. In that case, $\lambda_{\hat{H}_i}(\mathbf{u})$ is close to zero, which eliminates the influence of the Wiener-like filter in the new estimate $F_i(\mathbf{u})$ completely. $F_i(\mathbf{u})$ at the related frequencies is formed by the previous estimate $\hat{F}_i(\mathbf{u})$ which could be valuable due to the fact that its inverse Fourier transform satisfies the image space constraints. When the value of $|\hat{H}_i(\mathbf{u})|^2$ is comparable with β_i^h, $W_{\hat{H}_i}(\mathbf{u})$ is increasingly attenuated as $|\hat{H}_i(\mathbf{u})|^2$ decreases, and the value of $\lambda_{\hat{H}_i}(\mathbf{u})$ smoothly transfers from one to zero reducing the influence of $W_{\hat{H}_i}(\mathbf{u})G(\mathbf{u})$ on $F_i(\mathbf{u})$.

To make the interpolated Fourier space constraint in (5.7) work effectively, the two additional refinements described in §5.2.3.2 and §5.2.3.3 must be incorporated.

Experience indicates (Bates *et al.*, 1990a) that when either the image or the psf involved in the blurred image is symmetric (i.e. $f(\mathbf{x}) = f(-\mathbf{x})$), the performance of the algorithm is less effective than when it is applied to an asymmetric image, provided the the image is contaminated by the same level of SNR. The imaginary part of the spectrum of a symmetric image is zero. Thus if either component is known to be symmetric, the imaginary part of the corresponding spectral estimate can be set to zero as an additional Fourier space constraint.

5.2.3.2 Balancing the energy of $g(\mathbf{x})$ and its estimate

The consequence of energy bouncing between the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ (c.f. §5.1) is that the values of one estimate are significantly larger than what they should be, whilst the values of its counterpart at the related frequencies are distinctly smaller than they should be. The influence of the energy bouncing becomes worse when the interpolation of the estimate in (5.7) is introduced. A new strategy of balancing the energy of the estimates was therefore devised:

- Normalize $\hat{f}_i(\mathbf{x})$ and $\hat{h}_{i-1}(\mathbf{x})$ to a maximum magnitude of unity;
- Scale the given blurred image $g(\mathbf{x})$ such that its energy is equal to the energy of $\hat{f}_i(\mathbf{x}) \odot \hat{h}_{i-1}(\mathbf{x})$.

The first step confines the values of the input estimates at each iteration, $\hat{h}_{i-1}(\mathbf{x})$ and $\hat{f}_i(\mathbf{x})$ satisfying the image space constraints, within a fixed range. The consequent estimates $h_i(\mathbf{x})$ and $f_i(\mathbf{x})$ are therefore generated from a similar range of data. This makes the weighted average of the two items of the Fourier space constraint in (5.7) meaningful. Another advantage is that it takes account of the fact that the sizes of the image and psf are usually different.

The second step matches the energy of $g(\mathbf{x})$ with the energy of its estimate, which prevents the magnitude of the spectral estimate $H_i(\mathbf{u})$ or $F_i(\mathbf{u})$ generated from the same spectrum $G(\mathbf{u})$ to be distinctively different from what they should be. The combination of the two steps balances the energy and the range of the estimates. The performance of the algorithm is then expected to be more stable.

5.2.3.3 Scaling β_i

It has been discussed in §5.1 that a single filter constant β in the Davey Algorithm represents two different quantities, $|C(\mathbf{u})/H(\mathbf{u})|^{n+2}$ and $|C(\mathbf{u})/F(\mathbf{u})|^{n+2}$ respectively. A scheme of employing two filter parameters β_i^h and β_i^f (*cf.* Figure 5.1) in the AIA is proposed here. Since β_i^h affects the proportion of the effective and ineffective components of $\hat{H}_i(\mathbf{u})$ when performing the Wiener-like filter in (5.8), and is a quantity relative to $|\hat{H}_i(\mathbf{u})|^2$, it is therefore sensible to scale β_i^h by the maximum $|\hat{H}_i(\mathbf{u})|^2$, denoted $|\hat{H}_{i,max}(\mathbf{u})|^2$. β_i^f is similarly scaled by $|F_{i,max}(\mathbf{u})|^2$. Thus

$$\beta_i^h = \beta_i |\hat{H}_{i,max}(\mathbf{u})|^2, \tag{5.10}$$

and

$$\beta_i^f = \beta_i |\hat{F}_{i,max}(\mathbf{u})|^2. \tag{5.11}$$

Equations (5.10) and (5.11) show that β_i is a relative quantity for both β_i^h and β_i^f, the root square of which represents the proportion of $|\hat{F}_{i,max}(\mathbf{u})|$ and $|\hat{H}_{i,max}(\mathbf{u})|$ to be classified as ineffective parts of $\hat{F}_i(\mathbf{u})$ and $\hat{H}_i(\mathbf{u})$. Dynamically changing β_i discussed in

§5.2.2 actually varies the proportion of $|F_{i,max}(\mathbf{u})|$ and $|\hat{H}_{i,max}(\mathbf{u})|$ being effective and ineffective.

5.2.4 Computational examples

In this section computational examples are presented for two test purposes: Firstly, to demonstrate the effect of each of the refinements in §5.2.3 to the convergence of the AIA; Secondly, to compare the performance of the AIA with that of the Davey Algorithm.

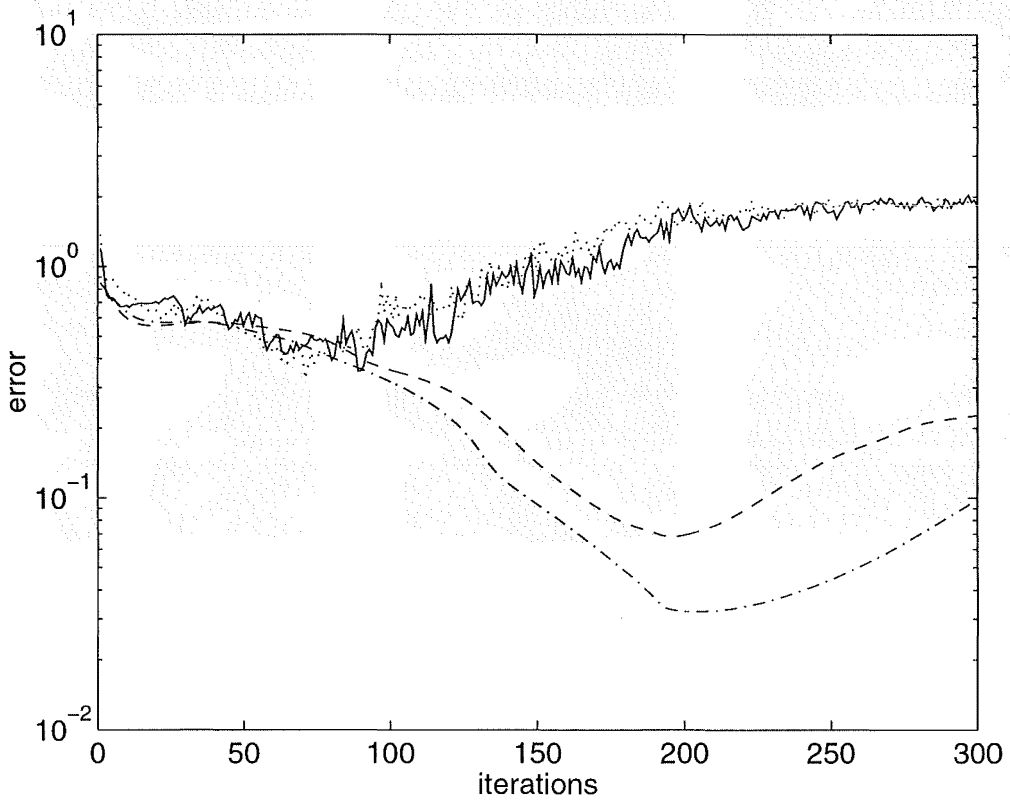


Figure 5.12 Examples of the effect of each of the refinements introduced in §5.2.3 on the convergence of the AIA, the AIA was applied on $g(\mathbf{x})$ shown in Figure 5.2(c) with SNR = 40dB, $k = 0.97, \beta = 0.1$, the related $f(\mathbf{x})$, $h(\mathbf{x})$ and $\hat{f}_0(\mathbf{x})$ are shown in Figure 5.2, true errors of the restorations, $E_T[f_i]$, obtained (a) including all refinements described in §5.2.3, depicted by dashdot curve; without including refinement of: (b) interpolation of $F_i(\mathbf{u})$ and $H_i(\mathbf{u})$, i.e. set $\lambda_{\hat{H}_i}(\mathbf{u}) = 1$ in (5.7), represented by continuous curve; (c) constraint for symmetric image, described by the dashed curve; (d) scaling β_i, depicted by the dotted curve.

Using the images displayed in Figures 5.2(a) to 5.2(e), the AIA was applied for the first test purpose with the iterative β_i reduction method to dynamically change β_i. Related parameters were selected as: $k = 0.97$, $\beta_0 = 0.1$ (*cf.* (5.1)), $I_D = 300$ (*cf.* step (vii) of §5.2.1) and $I_{max} = 300$. The algorithm was applied to $g(\mathbf{x})$ contaminated with SNR

of 40dB repeatedly, each time missed out one of the refinements described from §5.2.3.1 to §5.2.3.3. The resultant $E_T[f_i]$ in all cases are plotted in Figure 5.12.

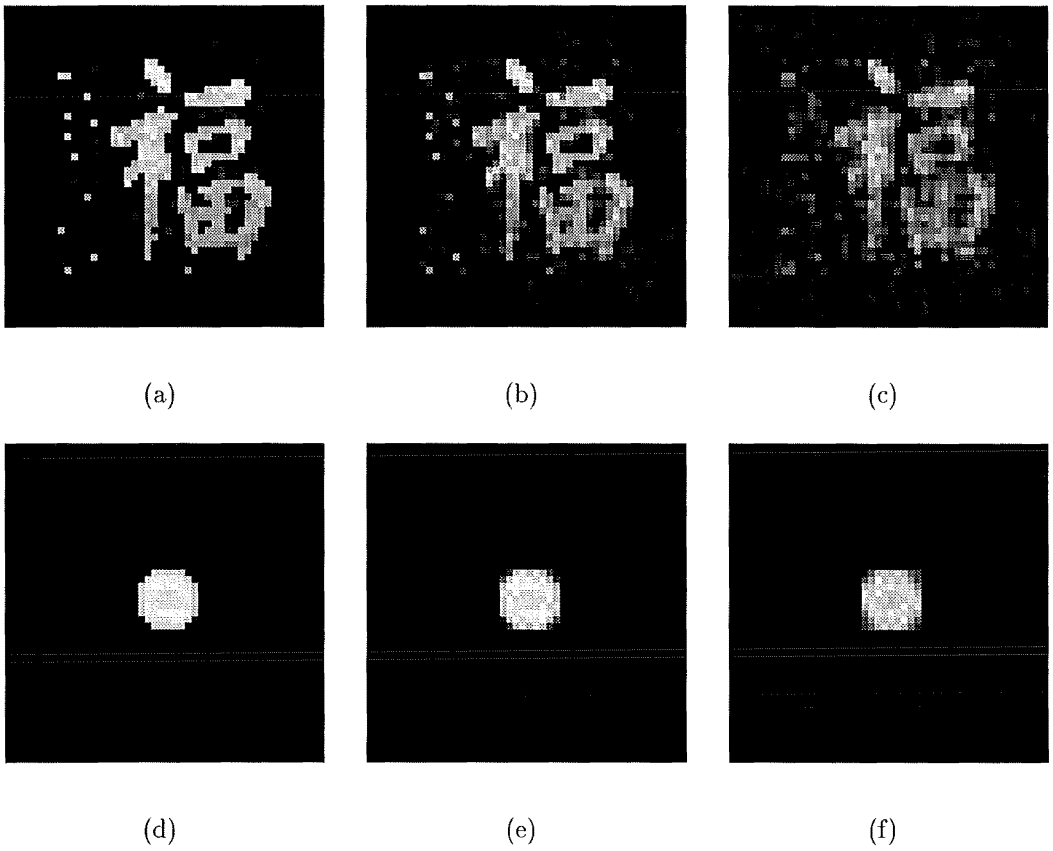


Figure 5.13 Comparative example of implementing the AIA, $\beta_0 = 0.1, k = 0.97$, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$, with $g(\mathbf{x})$ contaminated by : (a) and (d) SNR = 40dB; (b) and (e) SNR = 30dB; (c) and (f) SNR = 20dB.

For the second test purpose, both algorithms were applied to the same blurred image $g(\mathbf{x})$ with SNR = 40dB, 30dB and 20dB respectively. I_D is here selected 50 and the rest of the parameters used above for the AIA were employed here. To perform the Davey Algorithm, a group of fixed filter constants β were tested with $g(\mathbf{x})$ contaminated by each of the SNR levels. I_D and I_{max} were applied to the Davey Algorithm. The true error $E_T[f_i]$ was used to monitor the performance of the two algorithms. Restorations obtained from the optimal constant β_o, which resulted in the lowest $E_T[f_i]$, are used for comparison and are shown in Figure 5.14. Restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ using the AIA are presented in Figure 5.13. The true errors $E_T[f_i]$ in each cases are tabulated in Table 5.3.

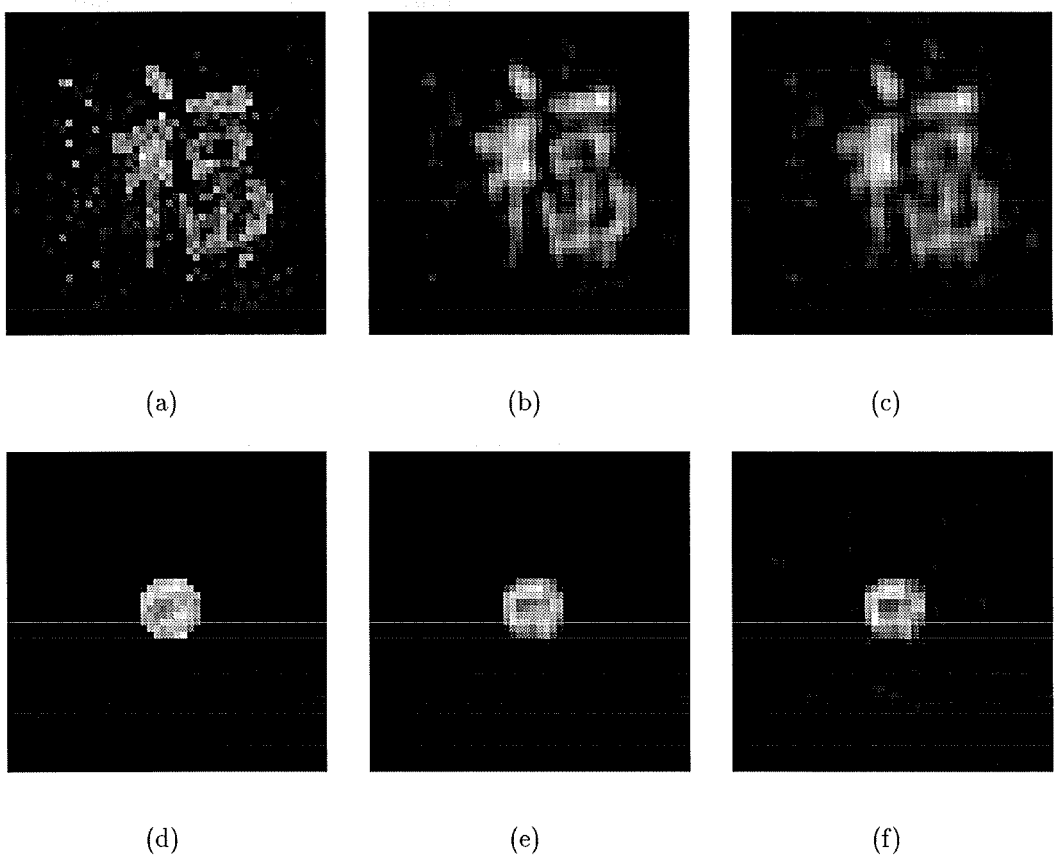


Figure 5.14 Comparative example of performing the Davey Algorithm using the optimal filter constant β_o, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$, separately, corresponding to Figures 5.2(a) to 5.2(c) with: (a) and (d) SNR = 40dB, $\beta_o = 10^{-9}$; (b) and (e) SNR = 30dB, $\beta_o = 5 \times 10^{-7}$; (c) and (f) SNR = 20dB, $\beta_o = 10^{-6}$

algorithm	SNR (dB)	$E_{T,min}[f_i]$	$E_T[h_i]$
AIA	40	0.032	0.014
	30	0.097	0.033
	20	0.247	0.064
Davey Algorithm	40	0.286	0.098
	30	0.283	0.132
	20	0.365	0.276

Table 5.3 Comparative example of implementing the AIA and the Davey Algorithm, corresponding to Figures 5.13 and 5.14.

5.2.5 Discussion

In this section the results presented in §5.2.4 and some aspects of the new algorithm are discussed.

Observation of the true error curves $E_T[f_i]$ in Figure 5.12 show that the minimum $E_T[f_i]$ achieved when using all the refinements of the AIA described in §5.2.3 is the lowest compared with those obtained without involving one of the refinements separately. The influence of the refinements of constraint for symmetric image, scaling β_i and the interpolated Fourier space constraint on the convergence of the algorithm increases in turn as shown by the corresponding minimum true error $E_T[f_i]$. The refinements presented in §5.2.3 are thus demonstrated to have improved the convergence of the algorithm, or equivalently, enhanced the accuracy of the restorations.

Comparison of the results in Figures 5.13 and 5.14 as well as their true errors $E_T[f_i]$ in Table 5.3 shows that the AIA retrieves restorations with better visual quality and lower error values than the Davey Algorithm at the same contamination level. The same conclusion can be made to the comparable results produced by the McCallum-Rodenburg Algorithm (*cf.* Figure 5.4). Concerning the computational cost when performing both algorithms, the Davey Algorithm must be performed for a number (typically 10) of different fixed constants β with 100 iterations for each β. In contrast, using appropriate β_i reduction rate and the initial filter parameter, typically $k = 0.97$ and $\beta_0 = 0.1$, the AIA can be routinely controlled, in terms of the filter parameter, satisfactory restorations can be produced straightaway despite the contamination level on $g(\mathbf{x})$ being unknown *a priori*. The performance of the AIA is therefore significantly accelerated compared with that of the Davey Algorithm.

It was noted in §5.2.2.4 that the size of the initial filter parameter β_0 does affect the convergence of the AIA. If β_0 is selected too small, for the given reduction rate k, the speed of recovering components of the restorations is slower than required by the decreasing rate of β_i. That is the effective components of $\hat{H}_i(\mathbf{u})$ set by β_i are not fully recovered or effective. When β_i reduces to an optimal value, from which β_i^h and β_i^f are the best estimates of their ideal quantities, such as shown in (4.58) for β_i^h, the restorations are not sufficiently recovered. In other words, components of $F(\mathbf{u})$ and $H(\mathbf{u})$ corresponding to the useful components of $G(\mathbf{u})$ are not mostly retrieved. Extra iterations are required to further retrieve the estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$. When β_i decreases to the value lower than the optimum, the contamination on $G(\mathbf{u})$ is likely to be overemphasized and the algorithm starts to diverge. If β_0 is selected to be smaller than the optimum β_i, no convergence of the algorithm would occur. An example is shown in Figures 5.11(c) and 5.11(d). Such behaviour of the AIA suggests that when the contamination level of the measurement is relatively higher, e.g. SNR = 20dB or worse, a larger β_0 may help to achieve better convergence of the algorithm since more iterations are therefore required to decrease β_i to its optimum. Experience shows that $\beta_0 = 0.1$ and $k = 0.97$ are good choice in general. However, when the blurred image is contaminated to a high level, such as SNR = 20dB or worse, $\beta_0 = 0.5$ and $k = 0.97$ or $k = 0.98$ may result in better convergence of the algorithm. In the examples demonstrated in chapter 6, for the first example involved with a bilevel image and contaminated $g(\mathbf{x})$ with

SNR of 20dB (see Figure 6.3), both pairs of $\beta_0 = 0.1$ and $k = 0.97$, and $\beta_0 = 0.5$ and $k = 0.97$ produce very similar results, while for the second example concerning a grey level image and $g(\mathbf{x})$ with SNR of 20dB (see Figure 6.7), the pair of $\beta_0 = 0.5$ and $k = 0.97$ produces lower true error than the pair of $\beta_0 = 0.1$ and $k = 0.97$ does. Note in chapter 6, only one pair of $\beta_0 = 0.1$ and $k = 0.97$ is used to produce all AIA results, in order to show the capability of the algorithm to produce satisfactory results with only one operation.

Each of the three methods of dynamically changing β_i presented in §5.2.2 has its advantages. The iterative β_i reduction approach in §5.2.2.2 is simple to perform. For large images or images contaminated with a high SNR level, such as SNR = 30dB or better, the method of adapting the rate of β_i reduction in §5.2.2.3 can further accelerate the performance of the algorithm compared with the former. The technique of adapting β_i to minimize the error metric makes the algorithm to be less independent on the suitability of the initial filter parameter β_0.

Results show that the convolutional error $E_a[g_i]$ shown in (4.67) correlates well with the true error $E_T[f_i]$. Experience also shows that the convolutional error $E_b[g_i]$ described in (4.67) often correlates better with $E_T[f_i]$ for the recommended reduction rate $k = 0.97$ and for the schemes proposed in §5.2.2.2 and §5.2.2.3.

5.3 AN ITERATIVE ALGORITHM FOR COLOURED NOISE

In this section a new iterative algorithm, the Coloured-noise Algorithm, to solve the blind deconvolution problem associated with coloured noise is presented. Apart from the Fourier space constraint, this algorithm is generally similar to the algorithm AIA presented in §5.2. The central part of the algorithm, the Wiener-like filter suitable for the coloured noise, is proposed in §5.3.1. The new algorithm is described in §5.3.2. Computational examples are given in §5.3.3 and results are discussed in §5.3.4.

5.3.1 A Wiener-like filter suited for coloured noise

Referring to the discussion in §4.2.2 regarding the "ideal" cut-off spectral magnitude $H_{max}(\mathbf{u})$ for the Wiener-like filter (4.3), $H_{max}(\mathbf{u})$ (4.57) is formed to prevent contamination domination in the new estimate $F_i(\mathbf{u})$. Since the spectrum of the convolution $G(\mathbf{u})$ used to derive $H_{max}(\mathbf{u})$ in (4.57) is represented by the true spectra $F(\mathbf{u})$ and $H(\mathbf{u})$ in (4.54), $H_{max}(\mathbf{u})$ is formed as a fixed function throughout the whole iterations.

An alternative approach to form a suitable cut-off spectral magnitude can be made by using $G(\mathbf{u})$ represented by the estimates of $F(\mathbf{u})$ and $H(\mathbf{u})$ at i^{th} iteration. Taking Fourier transform on both sides of (4.11)

$$G(\mathbf{u}) = \hat{F}_i(\mathbf{u})\hat{H}_i(\mathbf{u}) + C_i(\mathbf{u}), \tag{5.12}$$

and substituting for $G(\mathbf{u})$ in equation (4.53) (which is formed under the condition that $|\hat{H}_i(\mathbf{u})|$ is significantly greater than H_{max} and the Wiener-like filter (4.3) is reduced to an inverse filter), the estimate $F_i(\mathbf{u})$ becomes

$$F_i(\mathbf{u}) = \hat{F}_i(\mathbf{u}) + C_i(\mathbf{u})/\hat{H}_i(\mathbf{u}). \tag{5.13}$$

Although $\hat{F}_i(\mathbf{u})$, $\hat{H}_i(\mathbf{u})$ and $C_i(\mathbf{u})$ can be very poor estimates of $F(\mathbf{u})$, $H(\mathbf{u})$ and $C(\mathbf{u})$ respectively, especially at the early iterations, they show at least some resemblance. To prevent over exaggerating contamination in the restoration, it is sensible to prevent the contamination related term being appreciably greater than the signal related term in (5.13), i.e. to prevent

$$|\hat{F}_i(\mathbf{u})| \ll |C_i(\mathbf{u})/\hat{H}_i(\mathbf{u})| \tag{5.14}$$

on the RHS of (5.13). In other words, the spectral components of $|\hat{H}_i(\mathbf{u})|$ for which

$$|\hat{H}_i(\mathbf{u})| \ll |C_i(\mathbf{u})/\hat{F}_i(\mathbf{u})| \tag{5.15}$$

should be excluded from the process of producing the new estimate (5.13) by an inverse filter. This purpose can be achieved by selecting a functional cut-off spectral magnitude of $\hat{H}_i(\mathbf{u})$ at i^{th} iteration to be

$$H_{imax}(\mathbf{u}) = a\left|\frac{C_i(\mathbf{u})}{\hat{F}_i(\mathbf{u})}\right|, \tag{5.16}$$

where $H_{imax}(\mathbf{u})$ is a function of frequency $\mathbf{u}$ and varies from iteration-to-iteration, and a is a real constant satisfying $0 < a \leq 1$. The value of a is chosen so that (5.14) and (5.15) are satisfied; good performance of the algorithm is then assured.

Similar to the relationship between H_{max} and β in (4.52), a functional noise-to-signal ratio $\beta_i^h(\mathbf{u})$ is related to $H_{imax}(\mathbf{u})$ in (5.16) by

$$\beta_i^h(\mathbf{u}) = \frac{|H_{imax}(\mathbf{u})|^{n+2}}{n+1}. \tag{5.17}$$

Substituting $H_{imax}(\mathbf{u})$ by (5.16) and setting the exponent n to zero,

$$\beta_i^h(\mathbf{u}) = a^2\left|\frac{C_i(\mathbf{u})}{\hat{F}_i(\mathbf{u})}\right|^2. \tag{5.18}$$

Although $\beta_i(\mathbf{u})$ is a function of $\mathbf{u}$, it is obtainable from the estimates at the i^{th} iteration. Substituting β in (4.3) with $\beta_i(\mathbf{u})$ in (5.18) and setting $n = 0$, a Wiener-like filter for the coloured noise is formed as

$$\tilde{W}_{\hat{H}_i}(\mathbf{u}) = \frac{\hat{H}_i^*(\mathbf{u})}{|\hat{H}_i(\mathbf{u})|^2 + \beta_i(\mathbf{u})}. \tag{5.19}$$

Replacing $\beta_i(\mathbf{u})$ by the RHS of (5.18), equation (5.19) becomes

$$\tilde{W}_{\hat{H}_i}(\mathbf{u}) = \frac{\hat{H}_i^*(\mathbf{u})}{|\hat{H}_i(\mathbf{u})|^2 + a^2|C_i(\mathbf{u})/\hat{F}_i(\mathbf{u})|^2}, \tag{5.20}$$

where $C_i(\mathbf{u})$ can be calculated from (5.12),

$$C_i(\mathbf{u}) = G(\mathbf{u}) - \hat{F}_i(\mathbf{u})\hat{H}_i(\mathbf{u}). \tag{5.21}$$

When $H_i(\mathbf{u})$ is to be generated, the Wiener-like filter $\tilde{W}_{\hat{F}_i}(\mathbf{u})$ is established in the similar manner as that of $\tilde{W}_{\hat{H}_i}(\mathbf{u})$. That is

$$\tilde{W}_{\hat{F}_i}(\mathbf{u}) = \frac{\hat{F}_i^*(\mathbf{u})}{|\hat{F}_i(\mathbf{u})|^2 + a^2|C_i'(\mathbf{u})/\hat{H}_{i-1}(\mathbf{u})|^2}, \tag{5.22}$$

where $C_i'(\mathbf{u})$ is given by

$$C_i'(\mathbf{u}) = G(\mathbf{u}) - \hat{F}_i(\mathbf{u})\hat{H}_{i-1}(\mathbf{u}). \tag{5.23}$$

The Wiener-like filters in (5.19) and (5.22) place no restrictions on the type of additive noise involved. Whether the noise is white or coloured, the filters ensure that throughout the whole iterative process the spectral components of the new estimate, e.g. $F_i(\mathbf{u}) = \tilde{W}_{\hat{H}_i}(\mathbf{u})G(\mathbf{u})$, are most likely to be either dominated by the signal related term (for which $|\hat{H}_i(\mathbf{u})| \geq H_{imax}(\mathbf{u})$), or suppressed to a small value when $|\hat{H}_i(\mathbf{u})|$ is very small and the inverse filter in the related region is error sensitive (for which $|\hat{H}_i(\mathbf{u})| < H_{imax}(\mathbf{u})$).

5.3.2 Algorithm details

The Coloured-noise Algorithm is presented in this section to deal with blind deconvolution problems related to the coloured noise.

The capability of the algorithm to cope with the coloured noise is realised by involving the Wiener-like filter developed in §5.3.1 as the Fourier space constraint of the algorithm. That is to generate the estimate $F_i(\mathbf{u})$ by

$$F_i(\mathbf{u}) = \tilde{W}_{\hat{H}_i}(\mathbf{u})G(\mathbf{u}), \tag{5.24}$$

where $\tilde{W}_{\hat{H}_i}(\mathbf{u})$ is described in (5.20). Similarly the estimate $H_i(\mathbf{u})$ is yielded by

$$H_i(\mathbf{u}) = \tilde{W}_{\hat{F}_i}(\mathbf{u})G(\mathbf{u}), \tag{5.25}$$

where $\tilde{W}_{\hat{F}_i}(\mathbf{u})$ is described in (5.22).

The Coloured-noise Algorithm is formed by replacing the Fourier space constraint (*cf.* (5.7)) in step (iv) of §5.2.1 by (5.24), and removing step (ii) of §5.2.1 regarding the filter constant; the remainder of the steps of the algorithm in §5.2.1 are retained.

5.3.3 Computational examples

Examples of implementing the Coloured-noise Algorithm are illustrated in this section. Results are compared with those employing the Davey Algorithm.

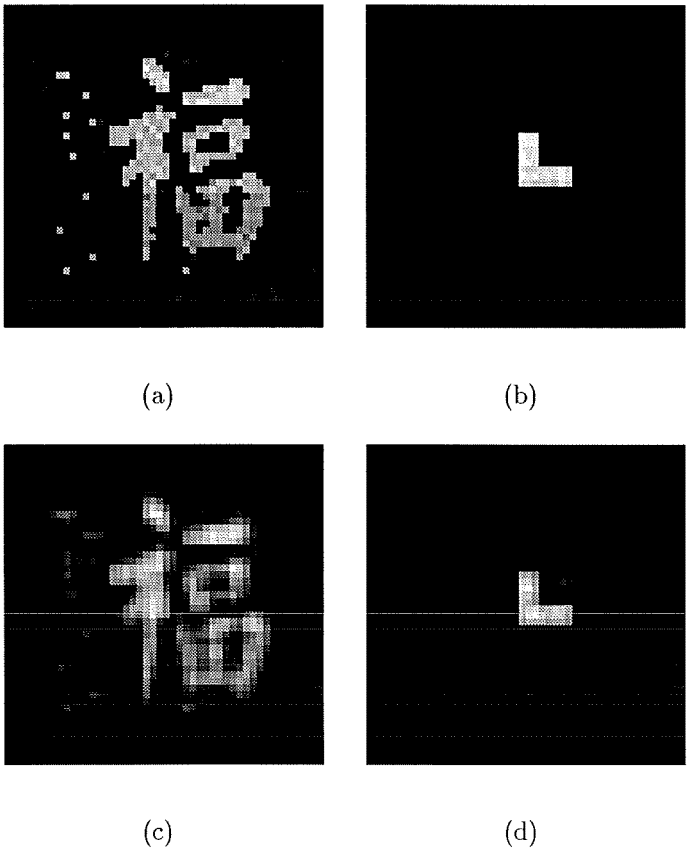
Both algorithms were respectively applied to a blurred image $g(\mathbf{x})$ formed by the convolution of the true image $f(\mathbf{x})$ shown in Figure 5.2(a) and a psf $h(\mathbf{x})$. $h(\mathbf{x})$ was "L" shape with 8×8 pixels in extent. $g(\mathbf{x})$ was contaminated by the additive coloured noise with SNR of 40dB. The coloured noise $c(\mathbf{x})$ was formed by filtering the power spectrum of the Gaussian noise by a highpass filter. Two pseudo-random images, shown in Figures 5.2(a) and 5.2(e), were used as the initial estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$. The image boxes of $f(\mathbf{x})$ and $h(\mathbf{x})$ were employed as the support estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ respectively. A maximal number of iterations is set to 300. The number of iterations between the one corresponding to the minimum $E_T[f_i]$ and the final iteration is set to be $I_D = 100$. Restorations were obtained at the iteration that $E_T[f_i]$ is minimum. When performing the Coloured-noise Algorithm, the constant a in (5.20) and (5.22) was selected to be 0.3 and the resultant restorations are shown in Figure 5.15. When implementing the Davey Algorithm, a group values of the filter constant β were tested. The one that resulted the lowest minimum $E_T[f_i]$ was selected and the corresponding restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ are displayed in Figures 5.15(c) and 5.15(d). The true error of the image in Figure 5.15(a) is $E_T[f_i] = 0.018$ and $E_T[f_i]$ of Figure 5.15(c) is 0.286.

5.3.4 Discussion

The results in Figure 5.15 demonstrate the capability of the Coloured-noise Algorithm to deconvolve images contaminated by the coloured noise. Comparison of the results produced by the Coloured-noise Algorithm and the Davey Algorithm in Figure 5.15 shows that better visual quality of the restorations and lower true errors $E_T[f_i]$ are obtained from the former. Besides, the Coloured-noise Algorithm is easier and faster to perform, in terms of the selection of the filter parameter, than the Davey Algorithm is. For the Coloured-noise Algorithm, there is no need to run the algorithm repeatedly to find a suitable values of β. Two main components $|C_i(\mathbf{u})|^2$ and $|\hat{F}_i(\mathbf{u})|^2$ of $\beta_i(\mathbf{u})$ (*cf.* (5.18)) are produced by the algorithm. Although the choice of the scalar "a" involved in $\beta_i(\mathbf{u})$ does affect the performance of the algorithm, a is much less sensitive to the contamination level of the blurred image than the filter constant β is. Experiments show that $a = 0.3$ has consistently resulted in satisfactory solutions for different contamination levels.

Experiments show that the convolutional error (*cf.* (4.66)) from the Coloured-noise Algorithm correlates well with the true error, however the behavior of this error metric is less stable than that of the Davey Algorithm. This is because of the involvement of the estimates of $F(\mathbf{u})$, $H(\mathbf{u})$ and $C(\mathbf{u})$ in the cut-off spectral magnitudes, or equivalently, $\beta_i(\mathbf{u})$. The effective and ineffective parts of $H_i(\mathbf{u})$ divided vary largely from frequency-to-frequency and from iteration-to-iteration.

It is worthwhile to emphasise that performing the same steps as depicted from

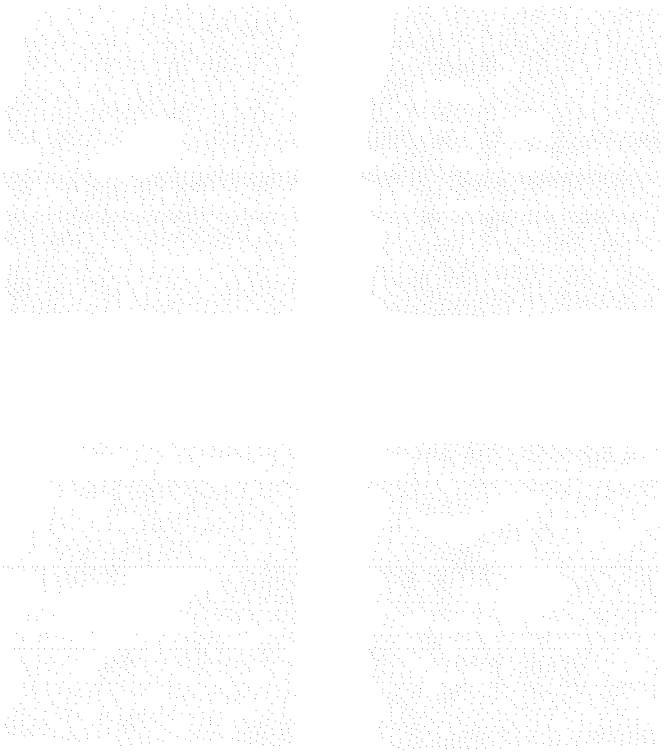


(a) (b)

(c) (d)

Figure 5.15 Comparative examples of blind deconvolution to the blurred image contaminated by the coloured noise with SNR = 40dB, restorations of $f(\mathbf{x})$, corresponding to Figure 5.2(a), and $h(\mathbf{x})$ obtained from: (a) and (b) the Coloured-noise Algorithm with $a = 0.3$; (c) and (d) the Davey Algorithm with the optimal filter constant $\beta_o = 10^{-6}$.

(5.12) to (5.19) and deriving the cut-off spectral magnitude of the standard Wiener filter in (3.9), the filter constant α of (3.9) can be related to the noise-to-signal ratio $|C(\mathbf{u})/F(\mathbf{u})|^2$ based on preventing contamination domination in the restoration. In this case, assumptions, such as that the noise is independent from the signal and has zero mean, are not required; the argument does not rely on the least squares optimality in that case. This may explain why the standard Wiener filter is still effective even when the assumptions to form the filter have not been satisfied, as mentioned by several authors (Bates *et al*, 1984; Seldin and Fienup, 1990).



Chapter 6

COMPARISON OF TWO BLIND DECONVOLUTION METHODS (AIA AND CGA)

Various blind deconvolution methods are reviewed in chapter 3. Two new methods are presented in chapter 5. Most of these methods, however, adopt different approaches to attack blind deconvolution problems. The comparative success of these methods for restoring images, measured in the quality of the restorations and the computational cost to obtain the restorations is little reported in literature. This information, however, is valuable when one is looking for some method to solve any particular blind deconvolution problem. Comparison of different methods on solving blind deconvolution problems is, therefore, the purpose of this chapter.

As discussed in chapter 3, blind deconvolution methods can be classified into three types: the iterative methods, the zero-based methods and the optimization methods. Considering the distinguishing features of each type of method, the iterative methods have the advantages of computational efficiency and ability to deconvolve images which are significantly contaminated. The zero based methods mentioned in §3.3 have shown their ability to deconvolve complex images (*cf.* Lane, 1988); however the methods are computationally expensive and only suitable for deconvolving uncontaminated images in present form. The class of optimization methods reviewed in §3.4 have the advantage of using theoretically well established optimization methods (McCallum, 1990b; Lane, 1992; Holmes, 1992) to approach blind deconvolution problems. These algorithms (for example, McCallum (1990b) and Lane (1992)) have demonstrated stable performance and the capability to deconvolve highly contaminated images.

Because of the limitation of zero based methods mentioned above, the comparison of different methods here is limited to iterative methods and optimization methods. Among three iterative algorithms reviewed in §3.2, the Davey Algorithm (Davey *et al.*, 1989; Davey, 1989; Bates *et al.*, 1990a) has demonstrated its superiority over the other two. Furthermore, the new algorithm presented in §5.2, the Automatic Iterative Algorithm (AIA), has achieved better results than the Davey Algorithm (see §5.2.4). The AIA is, therefore, selected to represent iterative methods for this comparison. Among optimization methods in §3.4, the Conjugate Gradient Algorithm (CGA) (Lane, 1992) is

chosen because it is less computationally intensive (compared with other optimization methods), capable of deconvolving considerably contaminated images and simple to operate. Comparison of blind deconvolution methods is thus narrowed down to the CGA and the AIA. The Conjugate Gradient Algorithm is described in §6.1. The comparative examples using two algorithms are presented in §6.2 and the results are discussed in §6.3.

6.1 CONJUGATE GRADIENT ALGORITHM DESCRIPTION

The AIA has been described in detail in §5.2 and the principle of the CGA has been described in §3.4.2. In this section, the steps of CGA, based on Polak-Ribiere method (Press *et al.*, 1992, §10.6), are presented. Reference is made to quantities of the error metric E_c and its components E_{sp} and E_{im} defined in (3.28), (3.29) and (3.30) respectively:

$$E_c = E_{sp} + E_{im}, \tag{6.1}$$

where

$$E_{sp} = \int_{\Theta^{sp}} |G(\mathbf{u}) - F_i(\mathbf{u})H_i(\mathbf{u})|^2 d\mathbf{u}, \tag{6.2}$$

and

$$E_{im} = \int_{\Omega_i^f} |f_i(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x} + \int_{\Omega_i^h} |h_i(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x}, \tag{6.3}$$

where Θ^{sp} represents the whole spectral domain, and Ω_i^f and Ω_i^h are the sets of pixels in $f_i(\mathbf{x})$ and $h_i(\mathbf{x})$, respectively, which violate the image space constraints.

To perform the CGA, two random images are generated and used as initial estimates of the image $f(\mathbf{x})$ and the psf $h(\mathbf{x})$, $\hat{f}_0(\mathbf{x})$ and $\hat{h}_0(\mathbf{x})$. An initial estimate of the composite image, fh_0, including both $\hat{f}_0(\mathbf{x})$ and $\hat{h}_0(\mathbf{x})$ is then formed.

(i) Set the initial minimum search direction $\mathbf{d}_0$ along the local downhill gradient $-\nabla E_c(p_0)$, i.e. the steepest descent direction, where p_0 is the starting point representing the initial estimate fh_0. The i^{th} searching direction is constructed by the vector

$$\mathbf{d}_i = -\nabla E_c(p_i) + \xi_{i-1}\mathbf{d}_{i-1}, \tag{6.4}$$

where the constant ξ_{i-1} is selected to make the new search direction $\mathbf{d}_i$ perpendicular to the old one $\mathbf{d}_{i-1}$, i.e. $\mathbf{d}_i \cdot \mathbf{d}_{i-1} = 0$. ξ_{i-1} is then given by

$$\xi_{i-1} = \frac{\nabla E_c(p_i)^T \nabla E_c(p_i)}{\nabla E_c(p_{i-1})^T \nabla E_c(p_{i-1})}, \tag{6.5}$$

where $\nabla E_c(p_i)$ is the gradient of $E_c(p_i)$ and $\nabla E_c(p_i)^T$ is the transpose of $\nabla E_c(p_i)$. Detail of calculating ∇E_c is presented by Lane (1992).

(ii) Generate the new composite image fh_i by

$$fh_{i+1} = fh_i + \Upsilon \mathbf{d}_i, \tag{6.6}$$

where Υ is a constant and is chosen to minimize the error metric $E_c(p_i)$ along the direction $\mathbf{d}_i$.

(iii) Update the estimates $f_{i+1}(\mathbf{x})$ and $h_{i+1}(\mathbf{x})$ from fh_{i+1}.

(iv) Steps (i) to (iii) form one direction search or one iteration. Repeat these steps until the difference of the error between the previous and current iterations $(E_c(p_{i-1}) - E_c(p_i))$ is smaller than a given value or the total number of iterations reaches the given maximum of 3000.

6.2 COMPARATIVE EXAMPLES

In this section, the AIA and the CGA are compared in terms of their ability to solve blind deconvolution problems. These algorithms were applied to the same measurements of blurred images using identical initial estimates $\hat{f}_0(\mathbf{x})$ and $\hat{h}_0(\mathbf{x})$. To avoid the possibility that one of the algorithms may perform better or worse on some particular type of measurement, four measurements, covering different types of image, psf and contamination levels, were used for this comparison. Each measurement was repeatedly employed by the AIA and the CGA, separately, to recover the estimates of the image and psf, each time starting with one of five different pairs of pseudo-random estimates $\hat{f}_0(\mathbf{x})$ and $\hat{h}_0(\mathbf{x})$. The image space constraints used for both algorithms were support and positivity. The best restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ for each algorithm, evaluated by the lowest observed error metric, were chosen for the comparison.

The first example comprises two measurements simulated by two versions (SNR $= \infty$ and 20dB) of convolution of a simple bilevel image $f(\mathbf{x})$ and an asymmetric psf $h(\mathbf{x})$ shown in Figures 6.1(a) and 6.1(b). The convolution $g(\mathbf{x})$ contaminated with SNR of 20dB is depicted in Figure 6.1(c). $f(\mathbf{x})$ and $h(\mathbf{x})$ are 32×32 and 8×8 pixels in extent in each coordinate direction, and are embedded in an array of 64×64 pixels respectively. The CGA and the AIA were applied to each of the measurements respectively, starting with a pair of random estimates $\hat{f}_0(\mathbf{x})$ and $\hat{h}_0(\mathbf{x})$. $\hat{f}_0(\mathbf{x})$ and $\hat{h}_0(\mathbf{x})$, shown in Figures 6.1(d) and 6.1(e), were constituted by pseudo-random numbers uniformly distributed between 0 and 1 within the support estimates $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$, separately. Five different pairs of $\hat{f}_0(\mathbf{x})$ and $\hat{h}_0(\mathbf{x})$ are formed in a similar manner but using different random seeds. $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$ are chosen to be equal to the image-boxes of $f(\mathbf{x})$ and $h(\mathbf{x})$.

The best restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ using the CGA were obtained at the maximum iteration of 3000 and are shown in Figures 6.2(a) and 6.2(b) for $g(\mathbf{x})$ uncontaminated and in Figures 6.3(a) and 6.3(b) for $g(\mathbf{x})$ contaminated with SNR of 20dB. The error

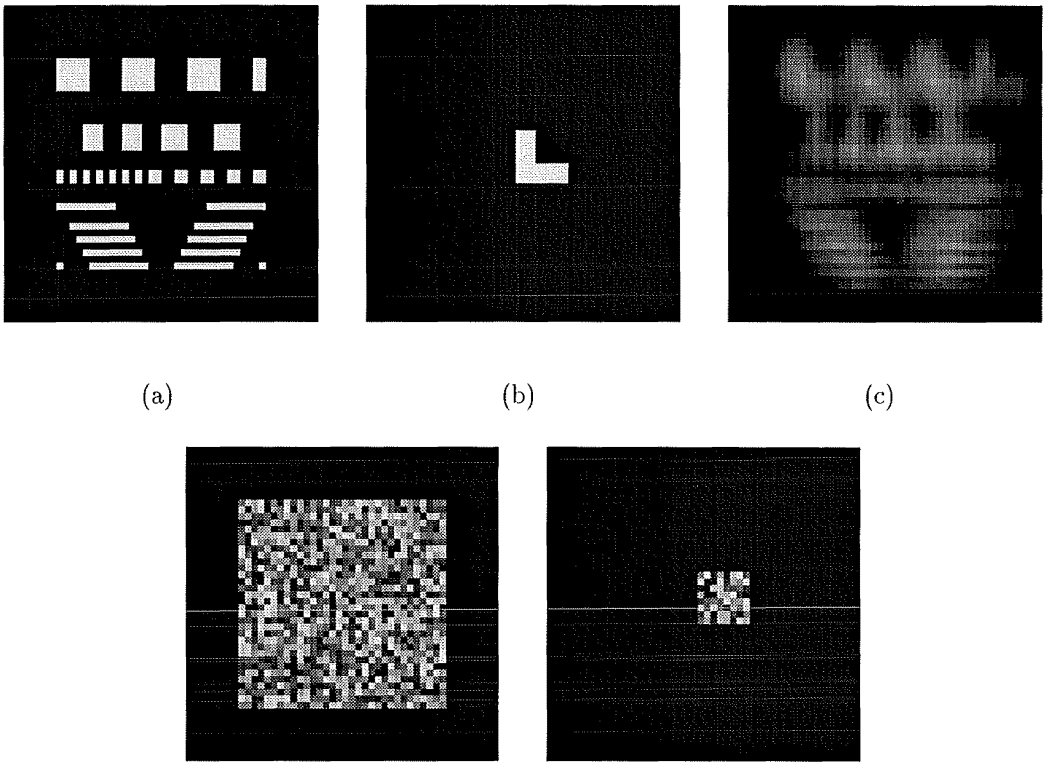
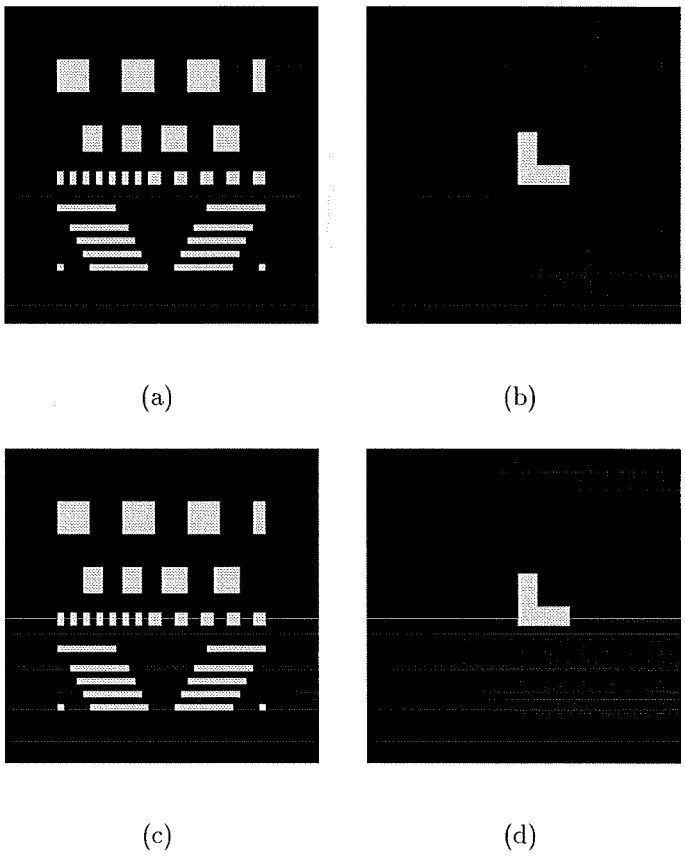


Figure 6.1 (a) A real positive image $f(\mathbf{x})$; (b) psf $h(\mathbf{x})$; (c) blurred image $g(\mathbf{x}) = f(\mathbf{x}) \odot h(\mathbf{x}) + c(\mathbf{x})$ with SNR = 20dB; (d) initial estimate $\hat{f}_0(\mathbf{x})$ and (e) initial psf estimate $\hat{h}_0(\mathbf{x})$.

curves E_c corresponding to five pairs of initial estimates for both convolution SNR levels are shown in Figures 6.4(a) and 6.4(c). The AIA was implemented using the iterative β_i reduction scheme described in §5.2.2.2. The convolutional error, $E_a[g_i]$ (4.67), was employed as error metric to indicate the "optimal" restorations and terminate the algorithm. Values of the related parameters were selected by $k = 0.97, \beta_0 = 0.1, I_{max} = 1000$ and $I_D = 100$. The best restorations from five deconvolutions (each using different starting images) are displayed in Figures 6.2(c) and 6.2(d) for $g(\mathbf{x})$ uncontaminated, and in Figures 6.3(c) and 6.3(d) for $g(\mathbf{x})$ contaminated with SNR of 20dB. All error curves of $E_a[g_i]$ in the two cases are plotted in Figures 6.4(b) and 6.4(d).

Since the error metric E_c from the CGA and $E_a[g_i]$ from the AIA are based on two different measurements and are incomparable, it is necessary to use some common error measure for numerical comparison of two algorithms. The true errors of $f(\mathbf{x})$ and $h(\mathbf{x})$, $E_T[f_i]$ in (5.3) and $E_T[h_i]$, are thus adopted for this purpose. $E_T[f_i]$ and $E_T[h_i]$ of the restorations in Figure 6.2 are tabulated in Table 6.1.

The second example involves two measurements $g(\mathbf{x})$ formed by convolving a grey scaled image and a symmetric psf corrupted with additive contamination of SNR of



(a) (b)

(c) (d)

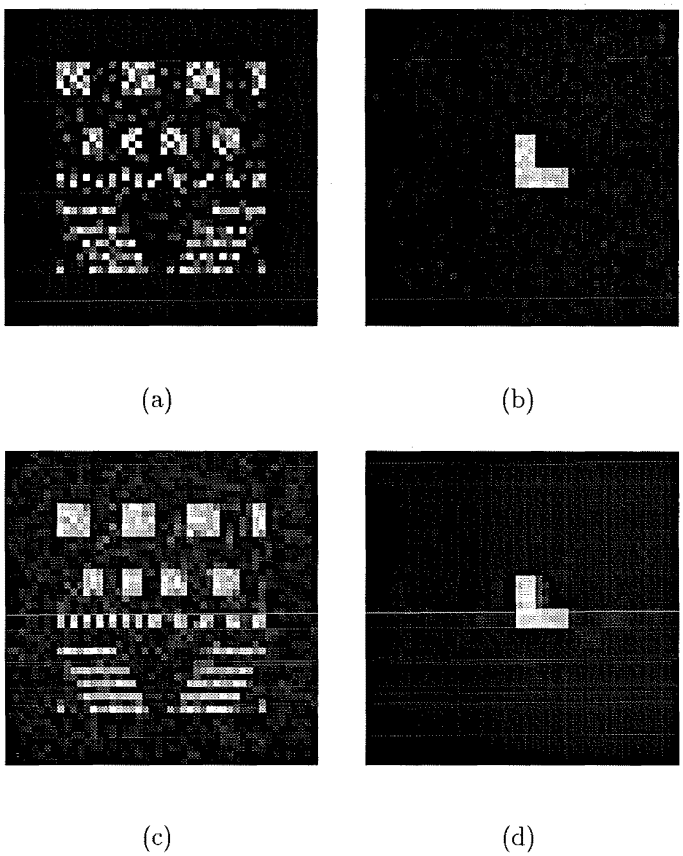
Figure 6.2 Example of comparative effectiveness of the CGA and the AIA in blind deconvolution, corresponding to Figure 6.1 with $g(\mathbf{x})$ uncontaminated, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ in Figures 6.1(a) and 6.1(b), separately: (a) and (b) produced by the CGA; (c) and (d) yielded by the AIA.

∞ and 20dB (shown in Figures 6.5(c)) respectively. The image $f(\mathbf{x})$ (shown in Figure 6.5(a)) is 32×32 pixels in extent and the psf $h(\mathbf{x})$ (shown in Figure 6.5(b)) is a disc with 7 pixels in diameter. Initial estimates $\hat{f}_0(\mathbf{x})$ and $\hat{h}_0(\mathbf{x})$, shown in Figures 6.5(a) and 6.5(b), were formed with pseudo-random values distributed within the correct support estimates $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$ separately.

When applying the CGA and the AIA to each version of the convolution in Figure 6.5(c), the same parameters as used in the first example were employed. The best restorations achieved by the CGA and the AIA when $g(\mathbf{x})$ is uncontaminated are shown in Figure 6.6. The best restorations obtained when $g(\mathbf{x})$ is contaminated with SNR of 20dB are displayed in Figure 6.7. The true errors $E_T[f_i]$ and $E_T[h_i]$ in the two cases are listed in Table 6.1.

6.3 DISCUSSION

Comparative examples illustrated in §6.2 are discussed in this section.



(a) (b)

(c) (d)

Figure 6.3 Example of comparative effectiveness of the CGA and the AIA in blind deconvolution, corresponding to Figure 6.1 with $g(\mathbf{x})$ contaminated with SNR of 20dB, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ in Figures 6.1(a) and 6.1(b), separately: (a) and (b) retrieved by the CGA; (c) and (d) obtained from the AIA.

corresponding to restorations in	true error | Iteration	algorithms	
		CGA	AIA
Figure 6.2	$E_T[f_i]\,\|I_T$	$2.04\times10^{-6}\|3000$	$1.88\times10^{-9}\|660$
	$E_T[h_i]\,\|I_T$	$2.81\times10^{-6}\|3000$	$9.45\times10^{-10}\|660$
Figure 6.3	$E_T[f_i]\,\|I_T$	$0.370\|3000$	$0.164\|142$
	$E_T[h_i]\,\|I_T$	$0.087\|3000$	$0.051\|142$
Figure 6.6	$E_T[f_i]\,\|I_T$	$7.83\times10^{-7}\|3000$	$0.033\|178$
	$E_T[h_i]\,\|I_T$	$1.98\times10^{-6}\|3000$	$0.054\|178$
Figure 6.7	$E_T[f_i]\,\|I_T$	$0.490\|3000$	$0.323\|129$
	$E_T[h_i]\,\|I_T$	$0.170\|3000$	$0.041\|129$

Table 6.1 Comparison of true errors of the restorations resulted by the CGA and the AIA. I_T represents the iteration number related to the true error on its left side.

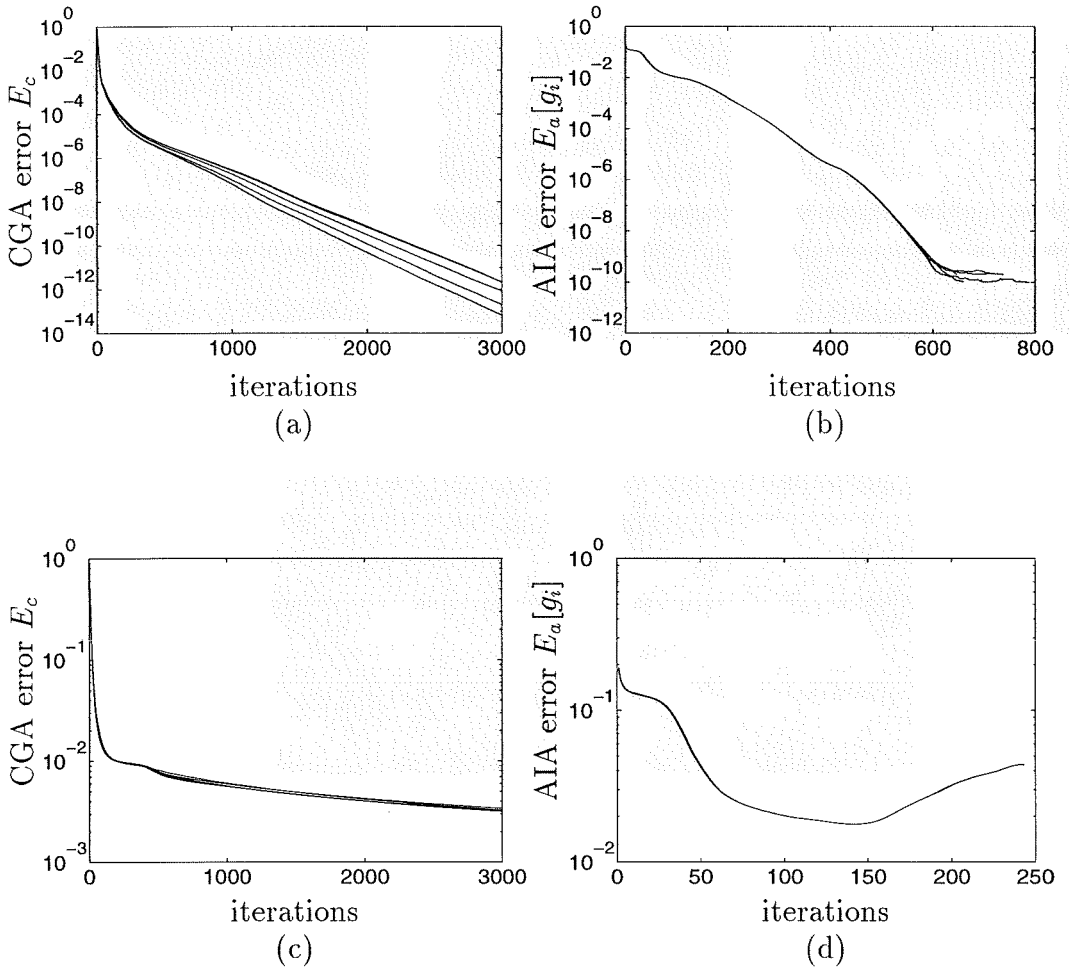


Figure 6.4 Demonstration of performance of the CGA and the AIA using five different pairs of starting random images, corresponding to Figures 6.2 and 6.3: (a) and (c) E_c curves resulting from the CGA when convolution SNR is ∞ and 20dB respectively; (b) and (d) $E_a[g_i]$ curves produced by the AIA when convolution SNR is ∞ and 20dB respectively.

Results of Figures 6.2 and 6.6 demonstrate that when the given blurred image $g(\mathbf{x})$ is uncontaminated, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ can be retrieved near perfectly using both algorithms, the CGA and the AIA. In the case that $g(\mathbf{x})$ is formed by a simple bilevel image and a "L" shaped psf (see Figure 6.1), although the true errors $E_T[f_i]$ and $E_T[h_i]$ of the restorations (see Table 6.1) from the AIA ($E_T[f_i] = 1.88 \times 10^{-9}$ and $E_T[h_i] = 9.45 \times 10^{-10}$) are lower than from the CGA ($E_T[f_i] = 2.04 \times 10^{-6}$ and $E_T[h_i] = 2.81 \times 10^{-6}$), the visual quality of the restorations resulting from two algorithms is identical. When $g(\mathbf{x})$ is generated from a more complicated image and a symmetric psf (see Figure 6.5), true errors ($E_T[f_i] = 7.83 \times 10^{-7}$ and $E_T[h_i] = 1.98 \times 10^{-6}$) from the CGA are much lower in values than those from the AIA ($E_T[f_i] = 0.033$ and $E_T[h_i] = 0.054$). However, as shown in Figures 6.6(a) and 6.6(c), the visual quality of the restorations of $f(\mathbf{x})$ from both techniques is not significantly different. The restoration of $h(\mathbf{x})$ from the CGA in Figure 6.6(b) does agree with its true version in Figure 6.5(b) better than that from the

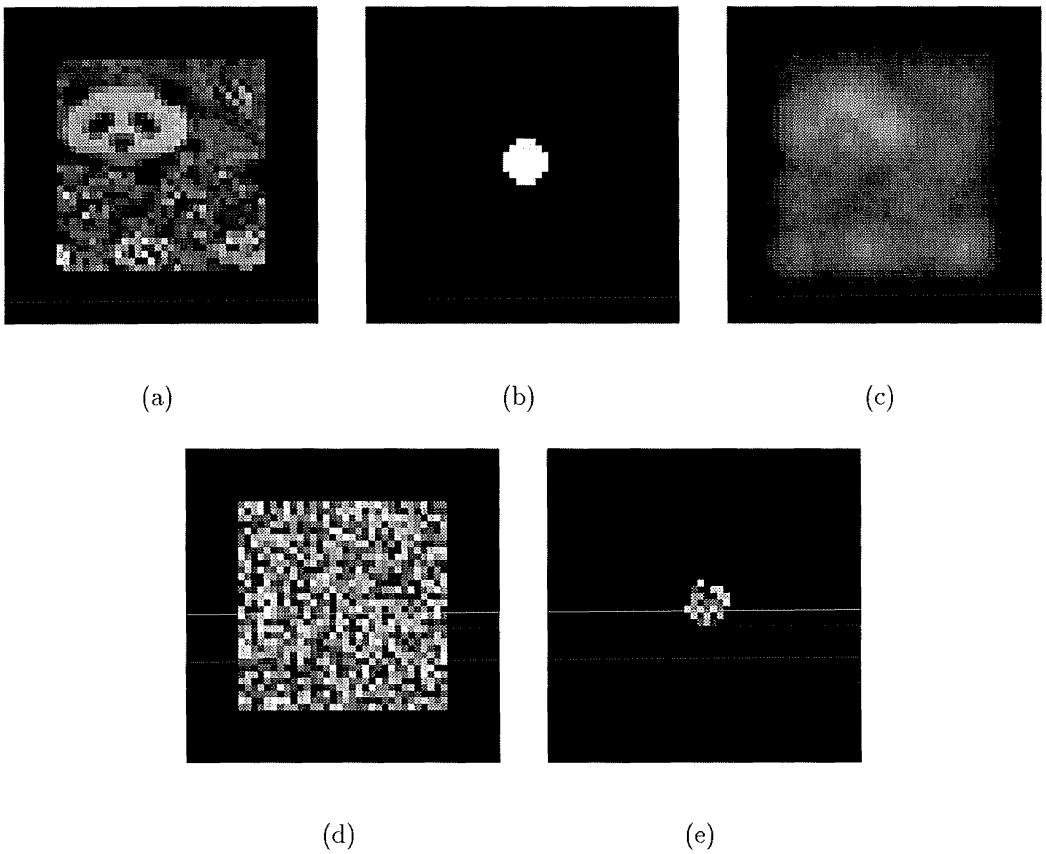


Figure 6.5 (a) A real positive image $f(\mathbf{x})$; (b) psf $h(\mathbf{x})$; (c) blurred image $g(\mathbf{x}) = f(\mathbf{x}) \odot h(\mathbf{x}) + c(\mathbf{x})$ with SNR = 20dB; (d) and (e) initial estimates $\hat{f}_0(\mathbf{x})$ and $\hat{h}_0(\mathbf{x})$ separately.

AIA in Figure 6.6(d).

Restorations in Figures 6.3 and 6.7 show that when the blurred images are contaminated (SNR = 20dB), the quality of the restorations retrieved from both algorithms of the CGA and the AIA are considerably degraded as expected. The visual quality of the psf restorations from the two algorithms are alike. However, the errors of the psf restorations (shown in Table 6.1) resulting from the AIA are smaller than those produced by the CGA. Considering the restorations of $f(\mathbf{x})$, the result in Figure 6.3(c) by the AIA demonstrates sound agreement with its true version of the bilevel image in Figure 6.1(a). Adding the same amount of convolution contamination and using the same algorithm AIA, the restoration of a grey level image in Figure 6.7(c) is noisier although the main feature of the image (the panda's head) is distinguishable. In contrast, restorations from the CGA (see Figures 6.3(a) and 6.7(a)) show less details of the images and are noisier than those from the AIA. However, restorations from the CGA are very clean outside of their supports, due to the application of the image space error E_{im} (6.3) involved in the error metric E_c (6.1). Evaluating the quality of these restorations by the true errors, results depicted in Table 6.1 show that true errors of the restored images yielded by the AIA are slightly lower than those from the CGA.

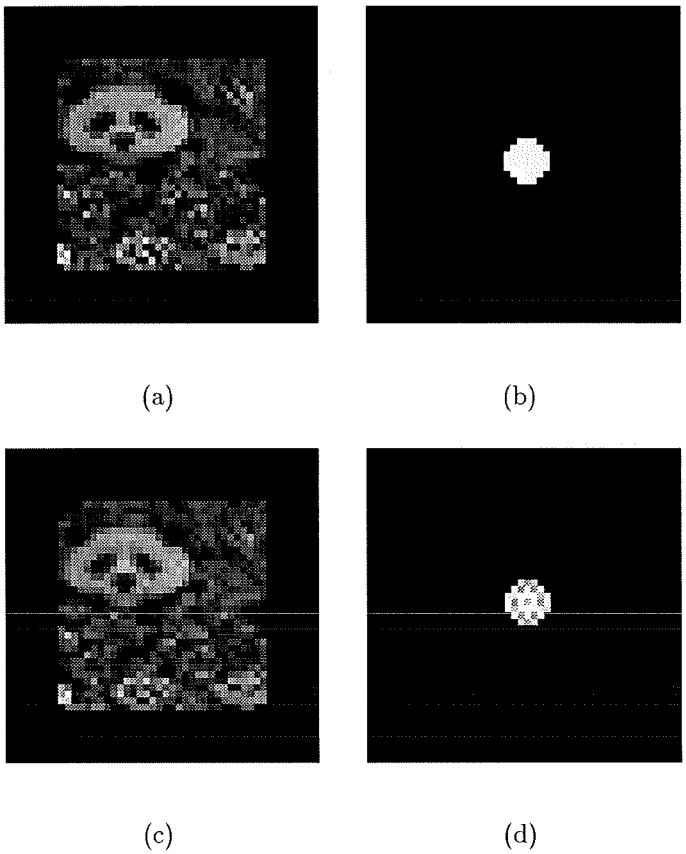


Figure 6.6 Second example of comparative effectiveness of the CGA and the AIA in blind deconvolution, corresponding to Figure 6.5 with $g(\mathbf{x})$ uncontaminated, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ in Figures 6.5(a) and 6.5(b), separately: (a) and (b) retrieved by the CGA; (c) and (d) resulting from the AIA.

In other words, the AIA results in better convergence of the algorithm than the CGA does.

It should be noted that all true errors $E_T[f_i]$ and $E_T[h_i]$ from the AIA were calculated before the constraints of support and positivity being applied. Inspection of the AIA restorations (e.g. Figures 6.3(c), 6.3(d) and 6.7(c)) shows that a considerable amount of contamination occurs outside of their supports. It is, therefore, expected that the values of these true errors will be further reduced if they are calculated after the constraints being enforced.

Inspection of the CGA restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ shows that the psf restoration is noiser outside of its support than the image restoration (see Figures 6.3(a) and 6.3(b), 6.7(a) and 6.7(b)). This suggests that the psf estimate does not influence the value of the image space error E_{im} as much as the image estimate does. The CGA, therefore, forces the two estimates to satisfy their image space constraints with very different weight. In other words, the CGA is not sensitive to the psf restoration in terms of how much it violates the image space constraints.

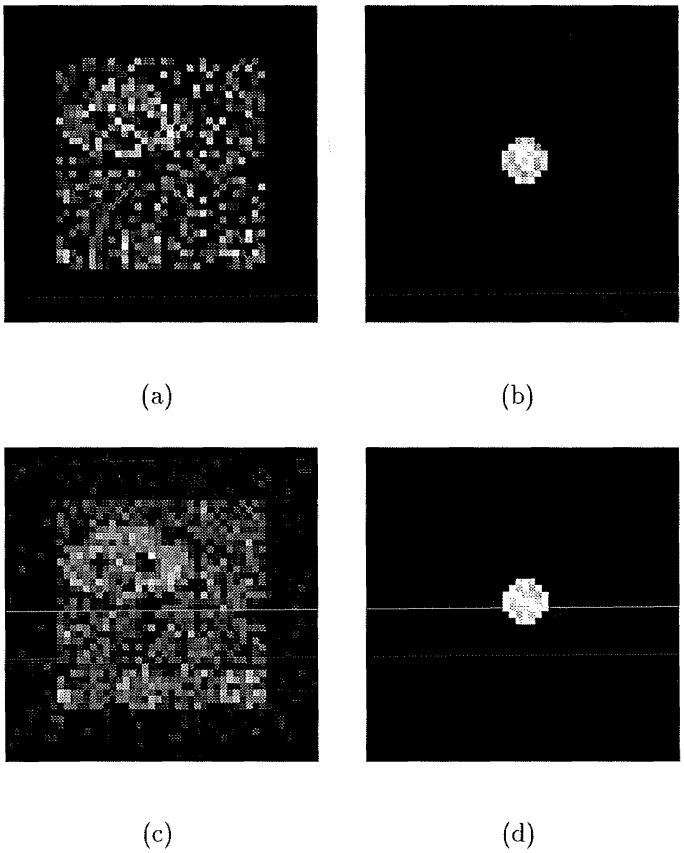


Figure 6.7 Second example of comparative effectiveness of the CGA and the AIA in blind deconvolution, corresponding to Figure 6.5 with $g(\mathbf{x})$ contaminated by SNR = 20dB, restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ in Figures 6.5(a) and 6.5(b), respectively: (a) and (b) obtained from the CGA; (c) and (d) restored by the AIA.

Yang *et al.* (1994) point out that the CGA is not scale invariant. Both pairs of $(\hat{f}, \hat{h})$ and $(\zeta\hat{f}, (1/\zeta)\hat{h})$, where ζ is a constant, yield the same data $g(\mathbf{x})$ and can be the solution of the algorithm. Different values of ζ results in different weights of the image estimate and the psf estimate in the error E_{im} and affect the performance of the algorithm differently. When ζ is large, the weight of the psf estimate in E_{im} is small. This is a case similar to the examples discussed in the previous paragraph.

Now consider the error metric behaviour in the algorithms (Figure 6.4). Little variation of the error curves (corresponding to five deconvolutions using different starting random images and psfs) occurs when $g(\mathbf{x})$ is contaminated with SNR of 20dB. There is a small variation in values of E_c and $E_a[g_i]$ when $g(\mathbf{x})$ is uncontaminated. The values of E_c (see Figures 6.4(a) and 6.4(c)) from the CGA decrease monotonically through all iterations, while the values of $E_a[g_i]$ (see Figures 6.4(b) and 6.4(d)) from the AIA fall continuously until reaching a minimum and then rise. Considering now the speed of the convergence of two algorithms, it took 660 iterations in the case with $g(\mathbf{x})$ uncontaminated and 142 iterations in the case with $g(\mathbf{x})$ contaminated for the AIA to reach the minima

of $E_a[g_i]$ (see Figures 6.4(b) and 6.4(d)), and produce the restorations in Figures 6.2(c) and 6.2(d), 6.3(c) and 6.3(d) respectively. In contrast, it took 3000 iterations in both cases of $g(\mathbf{x})$ uncontaminated and contaminated for the CGA to reach the level of E_c depicted in Figures 6.4(a) and 6.4(c), separately, and produce the restorations shown in Figures 6.2(a), 6.2(b), 6.3(a) and 6.3(b). Table 6.1 lists both the true errors of the restorations and the required iterations for all examples presented in §6.2. It took 4 times as many iterations for CGA to obtain the restorations displayed in Figure 6.2 than for AIA, and it took around 15 times as many iterations for CGA to generate the results show in Figures 6.3, 6.6 and 6.7 than for AIA. Correspondingly, it took the fastest 9 hours CPU time on SUN IPX and the slowest 19 hours CPU time on SUN IPC to produce the CGA results in Figures 6.2, 6.3, 6.6 and 6.7. In contrast, it took 1.5 hour CPU time on VAX 6000-430 hardware model type 161 to process the AIA results in Figure 6.2 and took maximum 0.5 hour CPU time on the same VAX to obtain the AIA results in Figures 6.3, 6.6 and 6.7.

When the AIA restorations in Figures 6.6(c) and 6.6(d) were produced, inspection of the error values of $E_T[f_i]$, $E_T[h_i]$ and $E_a[g_i]$ through all iterations shown that the minima of $E_T[f_i]$ and $E_a[g_i]$ correlated well, while minimum of $E_T[h_i]$ occurred 100 iterations after minima of $E_T[f_i]$ and $E_a[g_i]$ occurred. This observation shows that at the iteration the filter parameter β_i^h (see (5.8)) was reduced to the "optimum" value to produce minimum of $E_T[f_i]$, the filter parameter β_i^f at the same iteration was still not small enough to yield minimum of $E_T[h_i]$. Further reduction of β_i^f is, therefore, required while the "optimum" value of β_i^h is maintained. The restoration of $h(\mathbf{x})$ is then expected to be improved. Consequently, the restoration of $f(\mathbf{x})$ is anticipated to be also better in quality due to the improvement of the restoration of $h(\mathbf{x})$ and the suitable value of β_i^h and β_i^f used.

The more severe influence of convolution contamination on the performance of the CGA compared with that of the AIA (see Figures 6.3 and 6.7) is now discussed. For the CGA, the spectrum of $g(\mathbf{x})$, $G(\mathbf{u})$, is involved in one part of its error metric, E_{sp} described in (6.2), and is used to control or direct the formation of the new estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ at each direction search. The presence of contamination on $G(\mathbf{u})$ causes the error metric E_{sp} to deviate from the correct one. Thus, E_{sp} misleads the performance of the algorithm and the formation of the restorations. The contamination contained in $g(\mathbf{x})$ then directly affects the quality of the restorations. Furthermore, the involvement of the image space error E_{im} (6.3) in the error metric results in restorations having much smaller errors of E_{im} compared with those resulted from the AIA. It is argued that the effect of convolution contamination on the restorations may be focused within their supports because participation of E_{im} in the error metric guarantees at least one of two errors of the restorations of $f(\mathbf{x})$ and $h(\mathbf{x})$ outside of their supports to be considerably small. The algorithm is then more sensitive to the convolution contamination in terms of the quality of the resultant restorations within their supports. Suitability of the

involvement of E_{im} in the error metric in its present form is questioned thereby. When $g(\mathbf{x})$ is uncontaminated, E_{sp} is a true measure of convolution spectrum and direct the CGA to form a near perfect solution no matter the image is simple or complicated, as shown in Figures 6.2(a), 6.2(b), 6.6(a) and 6.6(b).

Unlike the CGA, the spectrum $G(\mathbf{u})$ in the AIA is involved in the Fourier space constraint to produce the new estimates of $f(\mathbf{x})$ and $h(\mathbf{x})$ at each iteration directly. The possible severe effect of convolution contamination on the estimates of $F(\mathbf{u})$ and $H(\mathbf{u})$ is attenuated by incorporating an interpolation of the samples of each of the estimates (see (5.7)). If at some frequencies $G(\mathbf{u})$ is likely to be dominated by contamination, it is designed that in such situation $G(\mathbf{u})$ is not involved to generate the new estimates. Instead, the new estimates at these frequencies are formed from their previous estimates at the same frequencies (see §5.2.3.1). The estimates of $F(\mathbf{u})$ and $H(\mathbf{u})$ resulting from the AIA are, therefore, not so heavily affected by the contamination on $G(\mathbf{u})$ compared with those produced by the CGA. In other words, the AIA is less sensitive to the presence of contamination.

In summary, the comparative examples of the performance of the CGA and the AIA show that the CGA is simple to perform. The algorithm has the advantage of being based on a theoretically based and proven optimization routine for its implementation. The performance of the algorithm is stable with monotonic decrease in errors. The main drawback of the algorithm, compared with the AIA, is that it is computationally expensive to operate. This is due to the nature of the optimization method in that the process of minimizing an image is carried out pixel by pixel rather than by treating the whole image for each calculation as the AIA does.

The AIA performs much faster than the CGA. Although the error metric of the AIA cannot be guaranteed in theory to decrease monotonically as the CGA can, in practice the error metric of the AIA decreases steadily until reaching a minimum. This minimum correlates reasonably well with that of the true error of $f(\mathbf{x})$ as demonstrated in §5.2.2.2, which makes it practical to terminate the algorithm appropriately. The AIA has the ability to reject contaminated samples of convolution spectrum and reduce the impact of convolution contamination on its restorations. Most of the simulation results in §6.2 show that the AIA is less sensitive to the presence of convolution contamination and produces better quality of the restorations compared with the CGA.

Chapter 7

REDUNDANCY IN MULTIDIMENSIONAL BLIND DECONVOLUTION

It has been discussed in §1.3 that the solution to the one-dimensional blind deconvolution problem is not unique. Given a blurred one-dimensional image, multiple solutions can be produced from the deconvolution process. However, a unique solution is almost always guaranteed when a two (or more) dimensional image is blindly deconvolved provided the true image and psf are compact (§2.2). In three (or more) dimensions, Millane (1990) has shown analytically that a unique solution can be obtained from fewer samples of the blurred image than are required by the Nyquist theorem. That is to say, a three (or more) dimensional image is overdetermined by its Nyquist samples. The amount of extra information contained in the image can be quantified in terms of a redundancy factor.

Three (or more) dimensional deconvolution has practical importance and occurs in situations relating to microscopy, seismology, radiology, crystallography, radar imagery and system identification (Conchello and Hansen, 1989; Carrington, 1990; Bose, 1990). It is therefore important to study the practical implication of Millane's theory for solving the multidimensional blind deconvolution problem. This chapter presents initial computational simulation on Millane's redundancy theory in multidimensional blind deconvolution. The preliminary results have been reported by Millane *et al.* (1994). The theory is discussed in §7.1. Two redundancy measures are introduced in §7.2. Computational simulations are presented in §7.3 and the results are discussed in §7.4.

Redundancy in multidimensional deconvolution is discussed for the three-dimensional (3D) case. However, results can be extended to higher dimensions in a straightforward manner.

7.1 THEORY

Consider an ideally blurred image $b(\mathbf{x})$ as described in (1.4). On replacing the position vector $\mathbf{x}$ by the Cartesian coordinates (x, y, z), a 3D blurred image $b(x, y, z)$ can be

described by

$$b(x,y,z) = f(x,y,z) \odot h(x,y,z). \tag{7.1}$$

The Fourier transform of (7.1) is

$$B(u,v,w) = F(u,v,w)H(u,v,w). \tag{7.2}$$

To achieve the unique blind deconvolution of a 2D image, its spectrum must be known continuously. In order for a digitized image to represent its continuous version uniquely, the sampling interval of the spectrum must be no greater than the Nyquist sampling interval. The uniqueness properties for 2D images can be straightforwardly applied to a 3D image $b(x,y,z)$, indicating that its Fourier transform $B(u,v,w)$ is required to be continuous for a unique solution. This requirement implies, according to the Nyquist sampling theorem, that $B(u,v,w)$ must be sampled at intervals of no more than $1/L_b(x)$, $1/L_b(y)$ and $1/L_b(z)$ in the u,v and w directions respectively, where $L_b(x), L_b(y)$ and $L_b(z)$ are the extents of the image box of $b(x,y,z)$ in the x,y and z directions. The extent $L_b(x)$ is related to the extents $L_f(x)$ and $L_h(x)$ according to the extent of convolution theorem described in (2.10), that is

$$L_b(x) \leq L_f(x) + L_h(x). \tag{7.3}$$

For discrete images, if $f(x,y,z)$ is real and positive, the following relationship holds:

$$L_b(x) = L_f(x) + L_h(x) - 1. \tag{7.4}$$

Relationships similar to (7.4) hold for both $L_b(y)$ and $L_b(z)$. It is shown in the following that the 3D blind deconvolution problem has a unique solution even if the Nyquist sampling requirements are relaxed (Millane, 1990b; Millane *et al.*, 1994). The description is presented in terms of Fourier space, so that $B(u,v,w)$ rather than $b(x,y,z)$ is considered.

Consider a 2D plane $B(u,v,\rho\triangle)$, where ρ is an integer and $\triangle$ represents the sampling interval in the w direction, let

$$B_\rho(u,v) = B(u,v,\rho\triangle). \tag{7.5}$$

From (7.2) and (7.5),

$$B_\rho(u,v) = F_\rho(u,v)H_\rho(u,v). \tag{7.6}$$

For each integer ρ, equation (7.6) poses a 2D blind deconvolution problem. The 3D blind deconvolution problem in (7.2) can therefore be treated by a set of 2D problems described by (7.6) with all possible values of ρ selected. Both $F_\rho(u,v)$ and $H_\rho(u,v)$ in (7.6) can be deconvolved or determined if $B_\rho(u,v)$ is available continuously. To obtain the continuous data from its digital version, the maximum sampling intervals of $B_\rho(u,v)$ must be no greater than the Nyquist spacing. These samples are $B(m/L_b(x), n/L_b(y), \rho\triangle)$ for all integers m,n. However, ambiguity exists with regard to the relative scale of the 2D

"image" $f_\rho(x,y) = IFFT\{F_\rho(u,v)\}$ and the "psf" $h_\rho(x,y) = IFFT\{H_\rho(u,v)\}$, and their positions. Blind deconvolution of $B_\rho(u,v)$ actually yields

$$\tilde{F}_\rho(u,v) = a_\rho F_\rho(u,v) e^{i2\pi(ux_\rho + vy_\rho)}, \tag{7.7}$$

and

$$\tilde{H}_\rho(u,v) = a_\rho^{-1} H_\rho(u,v) e^{-i2\pi(ux_\rho + vy_\rho)}, \tag{7.8}$$

where a_ρ, x_ρ and y_ρ are unknown constants. x_ρ and y_ρ represent the distances that $\tilde{f}_\rho(x,y) = IFFT\{\tilde{F}_\rho(u,v)\}$ and $\tilde{h}_\rho(x,y) = IFFT\{\tilde{H}_\rho(u,v)\}$ are translated in the x and y directions, respectively, in image space. In Fourier space, this translation becomes a linear phase shift in $\tilde{F}_\rho(u,v)$ and $\tilde{H}_\rho(u,v)$ as described by (7.7) and (7.8). Note that $\tilde{h}_\rho(x,y)$ and $\tilde{f}_\rho(x,y)$ are translated by equal distances but in opposite directions. The phase shift $e^{i2\pi(ux_\rho+vy_\rho)}$ of $\tilde{F}_\rho(u,v)$ cancels with the phase shift $e^{-i2\pi(ux_\rho+vy_\rho)}$ of $\tilde{H}_\rho(u,v)$ and thus $B_\rho(u,v)$ is given by

$$B_\rho(u,v) = \tilde{F}_\rho(u,v)\tilde{H}_\rho(u,v). \tag{7.9}$$

To build up $F(u,v,w)$ and $H(u,v,w)$ from $F_\rho(u,v)$ and $H_\rho(u,v)$, it is necessary to determine $F_\rho(u,v)$ from $\tilde{F}_\rho(u,v)$ and $H_\rho(u,v)$ from $\tilde{H}_\rho(u,v)$. To do this, the constants a_ρ, x_ρ and y_ρ must be determined. This can be achieved in the following manner.

The spectrum $B(0,v,w)$ can be considered to represent a 2D blind deconvolution problem in the same manner as above. From (7.5) and (7.6), $B(0,v,w)$ is given by

$$B(0,v,w) = F(0,v,w)H(0,v,w). \tag{7.10}$$

$F(0,v,w)$ and $H(0,v,w)$ can be blindly deconvolved using the samples $B(0, n/L_b(y), \rho/L_b(z))$. Following the same argument as used to obtain (7.7) and (7.8), the restored estimates of $F(0,v,w)$ and $H(0,v,w)$ can be described as

$$\hat{F}_0(v,w) = b_0 F(0,v,w) e^{i2\pi(vy_{u0} + wz_{u0})}, \tag{7.11}$$

and

$$\hat{H}_0(v,w) = b_0^{-1} H(0,v,w) e^{-i2\pi(vy_{u0} + wz_{u0})}, \tag{7.12}$$

where b_0, y_{u0} and z_{u0} are unknown constants. y_{u0} and z_{u0} represent the distances by which $\hat{f}_0(y,z) = IFFT\{\hat{F}_0(v,w)\}$ and $\hat{h}_0(y,z) = IFFT\{\hat{H}_0(v,w)\}$ are shifted in image space. The constants b_0, y_{u0} and z_{u0} can be set to arbitrary values. For each ρ, $\tilde{F}_\rho(u,v)$ and $\hat{F}_0(v,w)$ both incorporate the common line $(u=0, w=\rho\triangle)$. Therefore the constants a_ρ, x_ρ and y_ρ in (7.7) and (7.8) can be determined by equating $\tilde{F}_\rho(u,v)$ and $\hat{F}_0(v,w)$ along the line $(u=0, w=\rho\triangle)$. $F_\rho(u,v)$ and $H_\rho(u,v)$ are thereby determined.

To build up $F(u,v,w)$ continuously from $F_\rho(u,v)$ and $H(u,v,w)$ from $H_\rho(u,v)$, the sample spacing between any two adjacent $F_\rho(u,v)$ and $H_\rho(u,v)$ planes should be no greater than the Nyquist sampling interval, which is $1/L_f(z)$ and $1/L_h(z)$ respectively.

If there are enough samples for reconstructing the component ($F_\rho(u,v)$ or $H_\rho(u,v)$) with the larger extent, the same sampling interval is automatically small enough for reconstructing the other. Assuming $L_f(z) \geq L_h(z)$, the maximum sample spacing in the w direction that allows both $F(u,v,w)$ and $H(u,v,w)$ to be determined from $F_\rho(u,v)$ and $H_\rho(u,v)$ respectively is given by

$$\triangle = 1/L_f(z). \tag{7.13}$$

Therefore, the set of samples of $\{B(m/L_b(x), n/L_b(y), \rho/L_f(z)), B(0, n/L_b(y), \rho/L_b(z))\}$, for all integers m, n and ρ, are sufficient to uniquely solve the 3D blind deconvolution problem.

7.2 REDUNDANCY

The set of spectral samples described above that are sufficient to achieve a unique solution for the 3D blind deconvolution problem is smaller than the set $\{B(m/L_b(x), n/L_b(y), \rho/L_b(z))\}$ implied by a straightforward application of the Nyquist sampling theorem. Therefore the three (or more) dimensional blind deconvolution problem is said to be overdetermined. This overdeterminacy property can be quantified by various measures of redundancy (Millane, 1990b). One of these measures, denoted R_u, is given by the ratio of the number of samples implied by the Nyquist sampling theorem to the number of required samples based on the theory described in §7.1. In the 3D case,

$$R_u = \frac{L_b(x)L_b(y)L_b(z)}{L_b(x)L_b(y)L_f(z) + L_b(y)L_b(z)}. \tag{7.14}$$

Cancelling the term $L_b(y)$ out of (7.14),

$$R_u = \frac{L_b(x)L_b(z)}{L_b(x)L_f(z) + L_b(z)}, \tag{7.15}$$

which is greater than unity in general since $L_f(z)$ is usually smaller than $L_b(z)$. In the case of 2D blind deconvolution, both $L_b(z)$ and $L_f(z)$ are equal to unity, thus $R_u = 1$. There is no redundancy.

Another measure of redundancy is based on the number of Nyquist samples and denoted by R_n. R_n is the ratio of the number of Nyquist samples to the number of degrees of freedom (i.e. the number of Fourier coefficients) in the image. In the 3D case,

$$R_n = \frac{L_b(x)L_b(y)L_b(z)}{L_f(x)L_f(y)L_f(z) + L_h(x)L_h(y)L_h(z)}, \tag{7.16}$$

and in the 2D case,

$$R_n = \frac{L_b(x)L_b(y)}{L_f(x)L_f(y) + L_h(x)L_h(y)}. \tag{7.17}$$

If one of the convolution components is known, its extent at each coordinate direction can be measured, the extents of the other component can be found directly. The number

of degrees of freedom included in the denominator of (7.16) and (7.17) are therefore the maxima of the possible and R_n is then a conservative value.

Note that R_n does not equal unity in either the 3D or 2D case. Since the extent of the convolution is generally greater than the extent of each of its components, R_n is greater than unity, and the higher the dimensionality, the larger the redundancy R_n. R_n in 3D is greater than in 2D.

7.3 COMPUTATIONAL EXAMPLES

To study the practical implications of the oversampling theory, computational simulations using fully sampled and undersampled data, divided as schemes A, B and C, are performed. Results from 3D and 2D images are compared.

The blind deconvolution algorithm used for the simulation is modified from the Davey Algorithm (*cf.* §3.2.3), the filter constant β in (3.19) of which is replaced by a varied parameter β_i described in (5.1). Thus the Wiener-like filter becomes

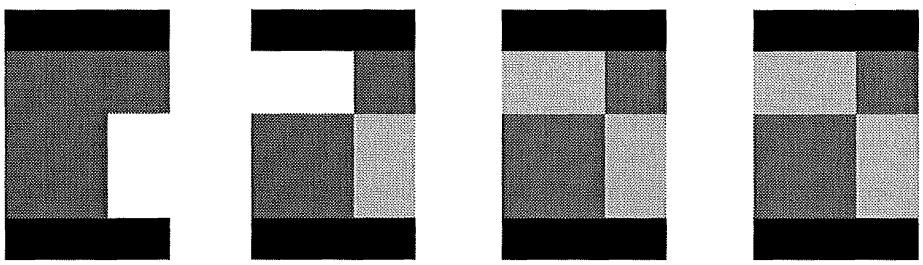
$$W_{\hat{H}_i}(\mathbf{u}) = \frac{\hat{H}_i^*(\mathbf{u})}{|\hat{H}_i(\mathbf{u})|^2 + \beta_i / \ |\hat{H}_i(\mathbf{u})|^n}. \tag{7.18}$$

The same notation describing the estimates of the image and psf in Figure 5.1 is employed here.

The algorithm is straightforwardly employed in the 3D and 2D simulations of the schemes A and B and incorporated with small modification in the scheme C. For all simulations, The parameters for the Wiener-like filter are selected as: $\beta_0 = 0.1$, $k = 0.8$ and $n = 0$. The blurred images are real, positive and uncontaminated. The initial estimates of the images are pseudo-random images and the correct supports of the image and psf are employed. The restorations presented are recorded at the iteration at which the minimum true error of the image estimate, $E_T[f_i]$, over 100 iterations occurs. The true error $E_T[f_i]$ is computed as described in (3.22). The only difference is that $E_T[f_i]$ here is calculated over the region within the support of the image rather than over the whole computed region.

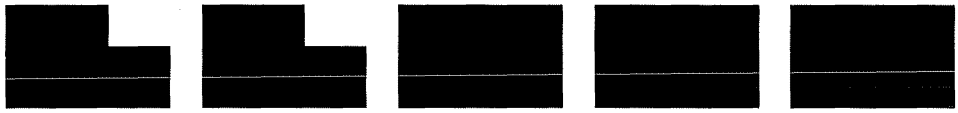
For the 3D simulation, the true image $f(x, y, z)$ is real and positive and has a size of $4 \times 12 \times 8$ voxels in extent in the x-y-z-directions. The four y-z-planes of $f(x, y, z)$ are displayed in Figure 7.1. The psf has a size of $5 \times 5 \times 8$ voxels and its five y-z-planes are shown in Figure 7.2. Both $f(x, y, z)$ and $h(x, y, z)$ were embedded in an array of $8 \times 16 \times 16$ voxels. The blurred image or the measurement $b(x, y, z)$, formed by the convolution of $f(x, y, z)$ and $h(x, y, z)$, is displayed in Figure 7.3. According to (7.4), the extents of $b(x, y, z)$ in the x-y-z-directions are $8 \times 16 \times 15$ voxels.

Referring to §7.1, the Nyquist sampling interval of the convolution spectrum $B(u, v, w)$ in the w direction is given by $1/L_b(z)$, while the maximum spacing that



x = 1 x = 2 x = 3 x = 4

Figure 7.1 Simulation of redundancy in 3D blind deconvolution: the four y-z-planes of the $4 \times 12 \times 8$-voxel true image $f(x, y, z)$. Only voxels within the support of the image and psf, or their restorations are displayed in the examples of this chapter. The y and z coordinate directions are shown vertically and horizontally.



x = 1 x = 2 x = 3 x = 4 x = 5

Figure 7.2 The five y-z-planes of the $5 \times 5 \times 8$-voxel psf $h(x, y, z)$.

allows unique deconvolution is $1/L_f(z)$. In the example presented here both $L_f(z)$ and $L_h(z)$ are equal to 8 voxels, $L_b(z)$ is equal to 15 voxels. Therefore the maximum spacing of $1/L_f(z)$ is nearly twice as large as the Nyquist sampling interval of $1/L_b(z)$, and the number of Nyquist samples is more than that minimally required. The fully sampled spectrum $B(u, v, w)$ with samples of $8 \times 16 \times 16$ voxels is therefore oversampled in the w direction according to the theorem discussed in §7.1. The redundancy measure R_u (see (7.15)) is equal to 1.52 for the simulation presented in this section and R_n (see (7.16)) is equal to 3.29. Restorations of $f(x, y, z)$ and $h(x, y, z)$ obtained from the fully sampled spectrum $B(u, v, w)$ are shown in Figures 7.4 and 7.5 respectively. A fully sampled spectrum is herein referred to as scheme A. The minimum true errors of $f(x, y, z)$ are listed in Table 7.1.

To test Millane's theory in §7.1 that a subset of Nyquist samples are sufficient to uniquely solve the blind deconvolution problem, the undersampled spectrum of the measurement, defined by $B_u(u, v, w)$, is simulated with two possible sampling schemes (B and C). $B_u(u, v, w)$ is equal to the fully sampled data $B(u, v, w)$ at every second u-v-plane and the central v-w-plane, and is zero elsewhere. Thus

$$B_u(u, v, w) = \begin{cases} F(u, v, w)H(u, v, w) & \text{for } w = 1, 3, 5, ..., N_z - 1, \\ F(u, v, w)H(u, v, w) & \text{for } u = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{7.19}$$

where N_z is the number of voxels in the z and w directions and is equal to 16. The

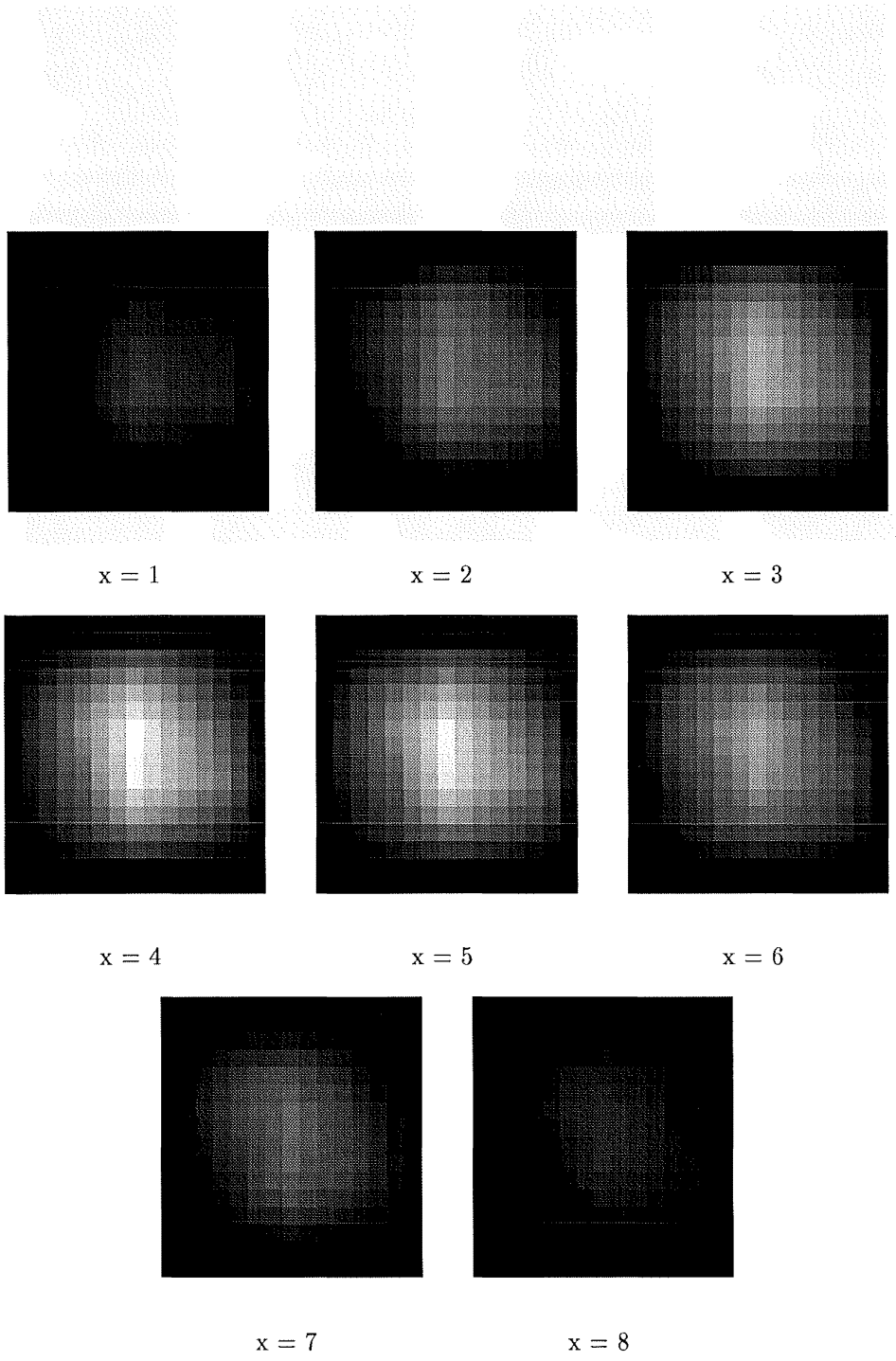


Figure 7.3 The eight y-z-planes of the $8 \times 16 \times 15$-voxels blurred image $b(x, y, z) = f(x, y, z) \odot h(x, y, z)$.

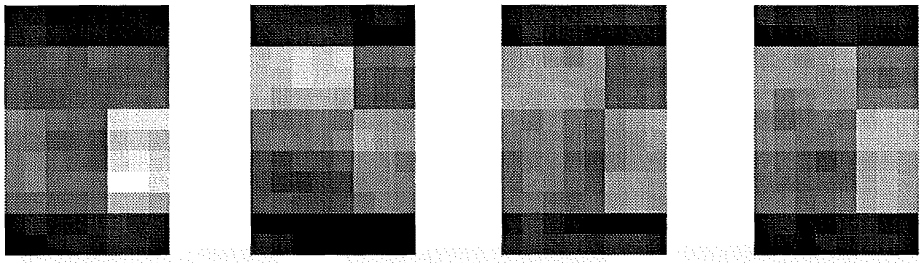


Figure 7.4 All y-z-planes of the restoration of the true image $f(x, y, z)$ shown in Figure 7.1. This restoration has been obtained from the fully sampled data $B(u, v, w)$ of scheme A.

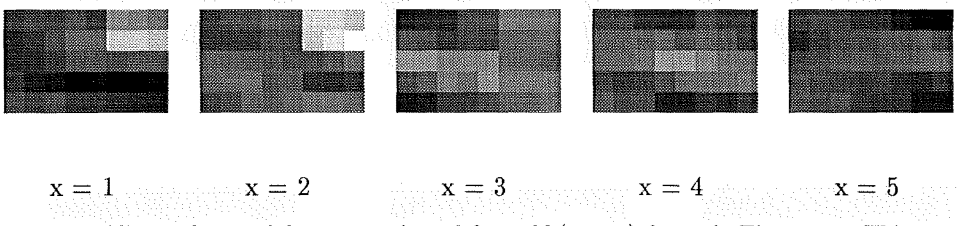


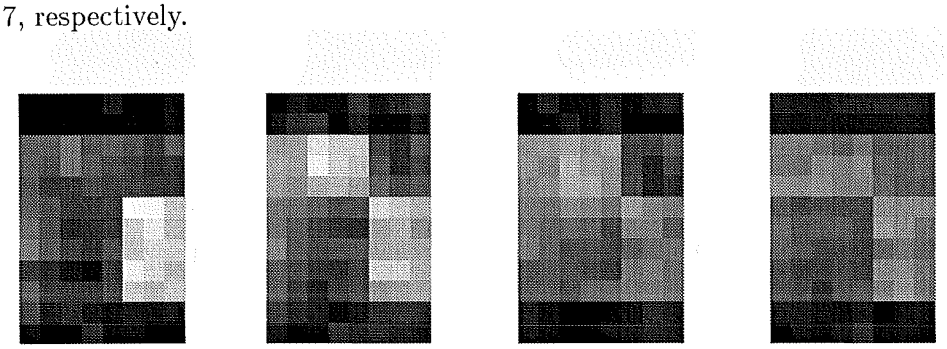
Figure 7.5 All y-z-planes of the restoration of the psf $h(x, y, z)$ shown in Figure 7.2. This restoration has been obtained from the fully sampled data $B(u, v, w)$ of scheme A.

sampling interval of $B_u(u, v, w)$ in the w direction is mostly equal to $2/N_z$, i.e. $1/L_f(z)$, except at frequencies where the central v-w-plane is located. Blind deconvolution using the undersampled data $B_u(u, v, w)$ can be performed in two ways. One method is to form the estimates of $F(u, v, w)$ and $H(u, v, w)$ which are sampled in the same manner as that of $B_u(u, v, w)$. That is the sample spacing of the estimates of $F(u, v, w)$ and $H(u, v, w)$ in each direction is the same as that of $B_u(u, v, w)$ (this shall be known as scheme B). The estimate of $F(u, v, w)$ generated at i^{th} iteration is written as

$$F_i(u, v, w) = W_{\hat{H}_i}(u, v, w) B_u(u, v, w), \tag{7.20}$$

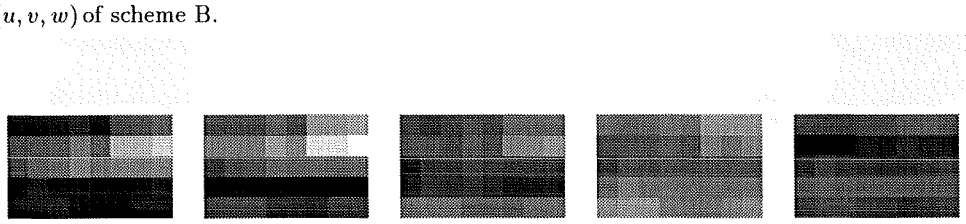
where $W_{\hat{H}_i}(u, v, w)$ is the Wiener-like filter described in (7.18). The effect of $F_i(u, v, w)$ being formed with sample spacing of $2/N_z = 1/L_f(z)$ in w is that $f_i(x, y, z)$ is repeated in the z direction with the period $L_f(z)$ over the range of $N_z = 2L_f(z)$ voxels. The implementation of the support constraint upon $f_i(x, y, z)$ produces the estimate $\hat{f}_{i+1}(x, y, z)$ and extends the period of $\hat{f}_{i+1}(x, y, z)$ in z to N_z voxels, which is twice as large as that of $f_i(x, y, z)$. Consequently, the spectrum of $\hat{f}_{i+1}(x, y, z)$, $\hat{F}_{i+1}(u, v, w)$, is an interpolated version of $F_i(u, v, w)$ and is fully sampled with sample spacing $1/N_z$ in w as that of $B(u, v, w)$. In other words, setting every second spectral sample of $\hat{F}_{i+1}(u, v, w)$ to zero produces a copy of $F_i(u, v, w)$ but only of half the magnitude. However, the interpolated samples of $\hat{F}_{i+1}(u, v, w)$ are not effectively used to generate the non-zero samples of $H_{i+1}(u, v, w)$, because the magnitude and phase of $B_u(u, v, w)$ at the related frequencies are equal to zero. The similar process occurs to $H_{i+1}(u, v, w)$. Restorations of $f(x, y, z)$ and $h(x, y, z)$ retrieved in this manner are shown in Figures 7.6

and 7.7, respectively.



x = 1 x = 2 x = 3 x = 4

Figure 7.6 All the y-z-planes of the restoration of $f(x, y, z)$ from the undersampled data $B_u(u, v, w)$ of scheme B.



x = 1 x = 2 x = 3 x = 4 x = 5

Figure 7.7 All the y-z-planes of the restoration of $h(x, y, z)$ obtained from the undersampled data $B_u(u, v, w)$ of scheme B.

Instead of forming the spectral estimates of $f(x, y, z)$ and $h(x, y, z)$, $F_i(u, v, w)$ and $H_i(u, v, w)$, with the same sample spacing as that of the convolution spectrum in scheme B, $F_i(u, v, w)$ and $H_i(u, v, w)$ are alternatively formed with sample spacing in w half as large as that of $B_u(u, v, w)$. The samples of $F_i(u, v, w)$ and $H_i(u, v, w)$ are computed from $B_u(u, v, w)$ at the planes where $B_u(u, v, w)$ has data, and are taken from their previous estimates, respectively, at the planes where $B_u(u, v, w)$ has no data (this undersampling of $B_u(u, v, w)$ is called scheme C), that is

$$F_i(u, v, w) = \begin{cases} W_{\hat{H}_i}(u, v, w) B_u(u, v, w) & \text{for } w = 1, 3, 5, ..., N_z - 1, \\ W_{\hat{H}_i}(u, v, w) B_u(u, v, w) & \text{for } u = 0, \\ \hat{F}_i(u, v, w) & \text{otherwise .} \end{cases} \tag{7.21}$$

The sample spacing of $F_i(u, v, w)$ and $H_i(u, v, w)$ in w is $1/N_z$ which is the same as the spacing of the fully sampled data $B(u, v, w)$ and half the spacing of the undersampled data $B_u(u, v, w)$. The restorations $F_i(u, v, w)$ and $H_i(u, v, w)$ from scheme C are presented in Figures 7.8 and 7.9.

For comparison, similar simulations of 2D images have been performed. The true image $f(x, y)$ and psf $h(x, y)$ (see Figures 7.10(a) and 7.10(b)) are formed by adding all the y-z-planes of the 3D $f(x, y, z)$ and $h(x, y, z)$, respectively, expanding each pixel by 2 in the y direction and by 4 in the z direction and extracting areas of 18×32 pixels

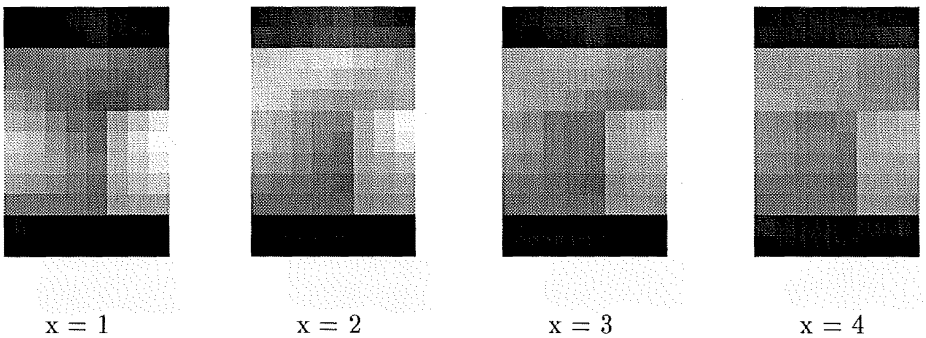


Figure 7.8 All y-z-planes of the restoration of $f(x, y, z)$ from the undersampled data $B_u(u, v, w)$ using scheme C.

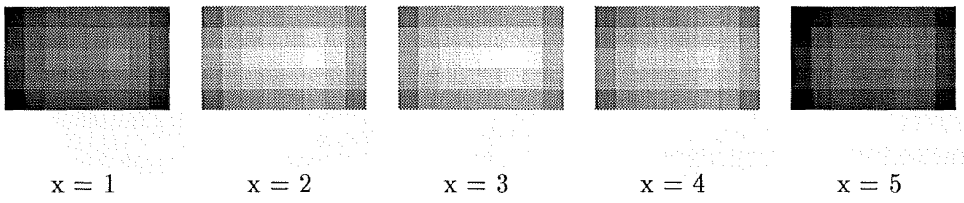


Figure 7.9 All y-z-planes of the restoration of $h(x, y, z)$ from the undersampled data $B_u(u, v, w)$ using scheme C.

for $f(x, y)$ and 15×32 pixels for $h(x, y)$. $f(x, y)$ and $h(x, y)$ are each embedded in an array of 32×64. The convolution of $f(x, y)$ and $h(x, y)$ (see Figure 7.10(c)) has 32×63 pixels which has a total number of data points similar to that of the 3D convolution $b(x, y, z)$. The redundancy measures R_u and R_n are shown in Table 7.2. Restorations of $f(x, y)$ and $h(x, y)$ using the fully sampled spectrum $B(u, v)$ are shown in Figures 7.11(a) and 7.11(b) (scheme A). The undersampled spectrum of the 2D measurement, denoted by $B_u(u, v)$, is formed by data in every second line of the fully sampled data $B(u, v)$ in the v-direction. Using $B_u(u, v)$, the blind deconvolution algorithm produces the spectral estimates of $F(u, v)$ and $H(u, v)$, $F_i(u, v)$ and $H_i(u, v)$, which are sampled in the same manner as $B_u(u, v)$. $F_i(u, v)$ and $H_i(u, v)$ have zero magnitude and phase at frequencies where $B_u(u, v)$ is zero (scheme B). Restorations of $f(x, y)$ and $h(x, y)$ from scheme B are presented in Figures 7.11(c) and 7.11(d). Corresponding to scheme C in the 3D case, $F_i(u, v)$ and $H_i(u, v)$ are generated from the undersampled data $B_u(u, v)$ in lines where $B_u(u, v)$ has information and preserved from their previous estimates otherwise. Restorations from scheme C are presented in Figures 7.11(e) and 7.11(f).

Inspection of 3D restorations of $f(x, y, z)$ and $h(x, y, z)$ in Figures 7.4 to 7.9 reveals that blind deconvolution is successful for both fully sampled and undersampled spectrum of the measurement. The most faithful restorations of $f(x, y, z)$ and $h(x, y, z)$ are obtained from the fully sampled data $B(u, v, w)$ and, as shown in Table 7.1, the true error of $f(x, y, z)$ is the lowest for scheme A. A considerable amount of the detail of $f(x, y, z)$ and $h(x, y, z)$ is retrieved from the undersampled data $B_u(u, v, w)$ using scheme

Scheme	Dimensions	$E_T[f_i]$	Iterations
A	3	0.019	75
	2	0.008	95
	2	0.029	75
B	3	0.030	69
	2	0.059	28
	3	0.053	28
C	3	0.043	14
	2	0.039	98
	2	0.062	14

Table 7.1 True errors of the restored images presented in Figures 7.4 to 7.11. Data in the top two rows of each scheme corresponds to the minimal errors.

Dimensions	R_u	R_n
3	1.52	3.29
2	1	1.9

Table 7.2 Redundancy in the 2D and 3D blind deconvolution example presented. The sizes of the 3D image (Figure 7.1) and psf (Figure 7.2) are $4 \times 12 \times 8$ and $5 \times 5 \times 8$. The sizes of the 2D image and psf (Figure 7.10) are 18×16 and 15×16.

B (see Figures 7.6 and 7.7). For scheme C, although the resolution of the restored image $f_i(x, y, z)$ (see Figure 7.8) is not as high as that of the estimates of $f(x, y, z)$ produced by the other two schemes, the basic structure of $f(x, y, z)$ appears in $f_i(x, y, z)$.

The 2D restorations $f_i(x, y)$ and $h_i(x, y)$ from the fully sampled data $B(u, v)$ (see Figures 7.11(a) and 7.11(d)) are very good and their accuracies are comparable to the corresponding 3D restorations. Blind deconvolution using undersampled data $B_u(u, v)$ is

(a) (b) (c)

Figure 7.10 Simulation of redundancy in 2D blind deconvolution: (a) true image $f(x, y)$; (b) psf $h(x, y)$; and (c) the blurred image $b(x, y) = f(x, y) \odot h(x, y)$.

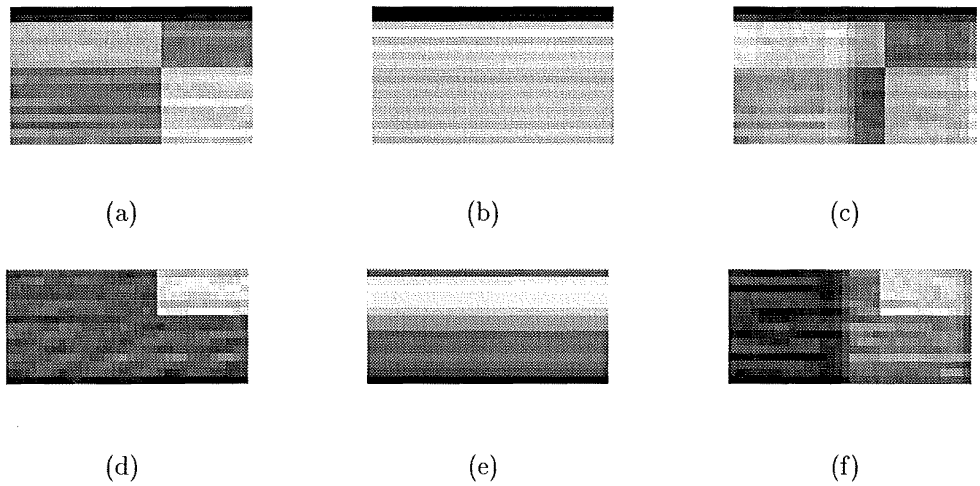


Figure 7.11 Restorations of $f(x, y)$ and $h(x, y)$ in Figure 7.10 from: (a), (d) fully sampled data $B(u, v)$; (b), (e) from undersampled data $B_u(u, v)$ of scheme B; (c) and (f) from undersampled data $B_u(u, v)$ of scheme C.

effective with scheme C (see Figures 7.11(c) and 7.11(f)) but failed with scheme B (see Figures 7.11(b) and 7.11(e)). The restorations $f_i(x, y)$ and $h_i(x, y)$ from the former scheme are better than those from the corresponding 3D scheme which are shown in Figures 7.8 and 7.9. However, $f_i(x, y)$ and $h_i(x, y)$ from the latter scheme do not even show the basic structure of their corresponding true images, especially in y direction.

Inspection of the minimum true errors of the restored images presented in the top two rows of each scheme in Table 7.1 reveals that for scheme B the lower error value resulted from the deconvolution of the 3D image. In contrast, for scheme A and C the lower error corresponds to the example of 2D deconvolution. However, comparing the true errors of the 3D to 2D estimates from the same iteration in each case, the lower error value always relates to the 3D case.

The redundancy measures, R_u and R_n, of the 3D and 2D examples are presented in Table 7.2. These measures show that higher redundancy corresponds to greater dimensionality; for the 3D example both R_u and R_n are over 1.5 times greater than the same redundancies calculated for the 2D example.

7.4 DISCUSSION

The simulation results for 3D and 2D blind deconvolution show that undersampling of the 3D convolution has rather small detrimental effect on the quality of the restorations compared with that using the fully sampled 3D data $B(u, v, w)$, while the effect of undersampling is more severe for the 2D case. Restorations obtained with undersampled 3D data $B_u(u, v, w)$ by Scheme B are much superior to those obtained with undersampled 2D data $B_u(u, v)$ by scheme B. The 2D restorations using scheme B show no sign of

recoverying in the direction corresponding to the undersampling of $B(u, v)$ and are much inferior than those retrieved from the fully sampled data. The superior performance of the undersampled 3D data to the 2D data is also indicated by the redundancy of the simulated images, R_u and R_n, both of which are over half times higher in 3D than in 2D. Therefore, experimental evidence supports the contention that the fully sampled 3D data $B(u, v, w)$, which satisfies the Nyquist sampling density, is oversampled, the 3D blind deconvolution problem is overdetermined, and the redundancy is more apparent in 3D than in 2D.

Possibly more significantly, the simulation results in Table 7.1 show that the 3D deconvolution algorithm appears to converge faster than the equivalent 2D case. It is important to note that because convergence with this iterative algorithm is variable, the results presented are not able to quantify precisely the improvement in convergence achieved in 3D compared to 2D.

The results also show that when using undersampled data with scheme C, the 3D restorations, especially the restored psf, are degraded compared to those obtained using scheme B. In contrast the 2D restorations of scheme C are significantly better than those of scheme B. A possible explanation for the superior performance of scheme B over scheme C in the 3D case could be that as the 3D undersampled data $B_u(u, v, w)$ is sufficient to retrieve $F_i(u, v, w)$ and $H_i(u, v, w)$, the combination of the samples of $F_i(u, v, w)$ and $H_i(u, v, w)$ generated from both $B_u(u, v, w)$ and their previous estimates (see (7.21) for $F_i(u, v, w)$) could introduce serious inconsistencies into their amplitudes in some cases by this algorithm as shown in this example and result in stagnation or less effective blind deconvolution. However, using the undersampled data $B_u(u, v)$ with scheme B in 2D case does not provide sufficient information to retrieve $F_i(u, v)$ and $H_i(u, v)$. Preserving samples from previous estimates at frequencies where the samples of $B_u(u, v)$ are missing actually (as in scheme C) compensates for the shortage of the samples of $B_u(u, v)$ to a certain extent. Although initially the amplitudes of the previous estimates of $F_i(u, v)$ and $H_i(u, v)$ are random, the influence of these amplitudes in the inverse Fourier transform of $F_i(u, v)$ and $H_i(u, v)$ is to produce pixels which violate image space constraints. Implementation of the image space constraints at each iteration gradually forces the amplitudes of these samples to converge to their solutions. The effect of the loss of information in $B_u(u, v)$ on $F_i(u, v)$ and $H_i(u, v)$ is therefore compensated for. The 2D restorations derived from undersampled data $B_u(u, v)$ with scheme C are unique and comparable to those obtained from the fully sampled data.

In applications, such as crystallography, where the image is periodic and 3D, its intensity measurement is sampled at twice the Nyquist spacing in each direction (Millane, 1990b). In such cases the redundancy apparent in 3D may play a significant role in achieving useful deconvolution and phase retrieval.

Chapter 8

CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

This chapter draws conclusions regarding original research presented in chapters 4 to 7. Several suggestions for further research in aspects of improving blind deconvolution techniques and simulation for redundancy in multidimensional blind deconvolution are made. The conclusions are presented in §8.1 and the suggestions for further research are discussed in §8.2.

8.1 CONCLUSIONS

Blind deconvolution of contaminated real or simple complex blurred images is now well known to be practical. A number of not well understood aspects of a leading iterative algorithm, the Davey Algorithm, have been extensively studied on a range of positive contaminated blurred images. Results show that the Wiener-like filter (*cf.* (4.3)) of the algorithm, when employed within the blind deconvolution loop, has a role distinct from that of a standard least squares filter (standard Wiener filter, *cf.* (3.6)). The Wiener-like filter is a modified inverse filter. It performs as an inverse filter when the involved function $\hat{H}_i(\mathbf{u})$ is effective (for which $|\hat{H}_i(\mathbf{u})| \geq H_{max}$) and is reduced to a small value when $\hat{H}_i(\mathbf{u})$ is ineffective (for which $|\hat{H}_i(\mathbf{u})| < H_{max}$). The proportion of effective and ineffective parts of $\hat{H}_i(\mathbf{u})$ is determined by the filter constant β which is an estimate of $|C(\mathbf{u})/F(\mathbf{u})|^{n+2}$ (*cf.* (4.58)) (when it is used to generate the estimate of $F(\mathbf{u})$). Both experiments and analysis show that the choice of β has significant influence on the performance of the Davey Algorithm and the quality of the restorations.

The integer n is related to the bias introduced by the Wiener-like filter. As n increases, the Wiener-like filter performs more accurately as an inverse filter in the effective part of $\hat{H}_i(\mathbf{u})$; and its magnitude is suppressed in a faster rate in the ineffective part of $\hat{H}_i(\mathbf{u})$. In other words, the larger n is, the less bias is involved in the filter. However, it is also noticed that a very large value of n may upset the smooth nature of the Wiener-like filter and therefore introduce artifacts in the restoration.

Comparison of the effect of the Wiener-like filter and the Seldin and Fienup filter

(*cf.* (4.62)) in the iterative blind deconvolution shows that the former produces slightly better restorations and much more stable performance of the algorithm.

The initial supports assumed for the image and psf, $M_f(\mathbf{x})$ and $M_h(\mathbf{x})$, are also critical for achieving optimal convergence of the algorithm and accurate restorations.

Good choices for the two critical parameters, β and $M_f(\mathbf{x})$ (or $M_h(\mathbf{x})$), are indicated by two different objective error metrics. The minimum of the convolutional error measure correlates well with a sensible choice of β; and the minimum of the image space error for the estimate of $h(\mathbf{x})$ indicates the best selection of $M_h(\mathbf{x})$. Consequently, a new method of estimating the best support estimate using the image space error is proposed.

Two new blind deconvolution iterative algorithms, the AIA and The Coloured-noise Algorithm, are presented in this thesis. The AIA not only has a reduced computational cost and little need for user intervention, but also produces more accurate restorations than the Davey as well as other algorithms. Furthermore, the performance of the AIA is stable and the termination point is well defined since the convolutional error correlates well with the true error of the restoration of $f(\mathbf{x})$.

The second algorithm is particularly suitable for deblurring images contaminated by coloured noise and results in better restorations than the Davey Algorithm. Like the AIA, this algorithm is straightforward to perform and converges faster, however, the performance of the algorithm shows less stability in its error metric than the AIA. As a by-product, it is pointed out that the standard Wiener filter can be derived from the sense of preventing contamination domination in the restoration. This may explain why the standard Wiener filter still works when the conditions to form this least squares filter are not met.

A comparison of two different types of blind deconvolution algorithms, the AIA and the CGA, shows that: the AIA, which has been demonstrated to be superior to other iterative algorithms, is much less computationally intensive and converges faster than the CGA, which is a representative optimization method; moreover, for different contaminated blurred images, the AIA produces better quality restorations. The comparison shows that the CGA, which uses an error metric to direct the convergence of the algorithm, is very sensitive to contamination in the blurred image due to ill-posed nature of the deconvolution problem.

Original simulations for 3D and 2D blind deconvolution show that using undersampled data (scheme B), effective 3D restorations are achieved, whilst the restorations in 2D fail in the direction corresponding to the undersampling. Two redundancy metrics for the simulated images are higher in 3D than in 2D. The study supports the contention that blind deconvolution in three or more dimensions is overdetermined by the Nyquist samples of the convolution.

8.2 SUGGESTIONS FOR FURTHER RESEARCH

8.2.1 Zero-based methods

As reviewed in §3.3, zero based deconvolution methods are less sensitive to the inaccuracy of the psf estimate than other methods since explicit use of the psf estimate in the restoration of the image estimate is avoided. Once the zero-sheet of $F(\mathbf{u})$ is separated from the zero-sheet of $G(\mathbf{u})$ with the aid of the zero-sheet of $H(\mathbf{u})$, the estimate of the image is restored directly from the zero-sheet of $F(\mathbf{u})$. It has been demonstrated that the zero methods are able to restore images with high accuracy provided only a modest amount of contamination is present (Watson *et al.*, 1992).

When $g(\mathbf{x})$ is contaminated, separation of zero-sheets of $F(\mathbf{u})$ and $H(\mathbf{u})$ from zero-sheet of $G(\mathbf{u})$ is more difficult. To take the advantage of the zero methods in such cases, one possible approach is to combine an iterative blind deconvolution algorithm, such as the AIA, with the zero-based method. In brief: AIA is first performed to produce estimates $\hat{f}(\mathbf{x})$ and $\hat{h}(\mathbf{x})$; the zero-sheets of $\hat{H}(\mathbf{u})$ and $G(\mathbf{u})$ are then calculated and the sections of the zero-sheet for $G(\mathbf{u})$ corresponding to $\hat{H}(\mathbf{u})$ are removed; the remainder is compared with the zero-sheet calculated from $\hat{F}(\mathbf{u})$ and matching portions are used to derive a new estimate for $f(\mathbf{x})$. Since direct use of the estimate $\hat{h}(\mathbf{x})$ is avoided in restoring the new estimate of $f(\mathbf{x})$, higher accuracy of this restoration may be expected. This kind of combination of an iterative method with zero methods has been used for phase retrieval and better convergence of the algorithm has been demonstrated (Parker and Bones, 1992b; Parker, 1994).

8.2.2 The iterative algorithm

One out of four examples demonstrated in chapter 6 indicates that the method of calculating the two filter parameters β_i^h and β_i^f in the AIA (*cf.* §5.2.3.3) is not always adequate and further reduction of β_i^f is required to achieve better convergence of the AIA.

One possible avenue is now described. Once the AIA (*cf.* §5.2) has converged (recall that the algorithm is halted if no lower error metric is reached I_D iterations after a local minimum), β_i^h can be held constant from the point of local minimum while β_i^f is allowed to decrease further. If the error metric is not further reduced by this process, the algorithm can be assumed to have converged.

8.2.3 Simulation of redundancy in 3D

Although the simulation results for 3D and 2D images presented in chapter 7 provide evidence to support the contention of redundancy in multidimensional blind deconvolution,

the iterative algorithm applied did not exhibit stable convergence. To show whether a unique solution always exists for the proposed 3D undersampled data (*cf.* §7.1) and the practical implication of this contention more convincingly, more advanced algorithms, in terms of stability and convergence, are necessary. The AIA presented in §5.2 may be a good choice especially for contaminated data. When the data is uncontaminated, such as in the simulation presented, the CGA (*cf.* §3.4.2) is more powerful. As has been mentioned in §6.3 that when the image is uncontaminated, the error metric described in (3.28) is a true measure which directs the algorithm to converge correctly, steadily and forcefully towards its solution. A range of further experiments employing the AIA for contaminated data and the CGA for uncontaminated data is therefore suggested.

REFERENCES

ABLES, J.G. (1974), 'Maximum entropy spectral analysis', *Astronomy and Astrophysics Supplement Series*, Vol. 15, No. 3, June, pp. 383–393.

ANDREWS, H.C. and HUNT, B.R. (1977), *Digital Image Restoration*, Prentice-Hall, New Jersey.

AYERS, G.R. and DAINTY, J.C. (1988), 'Iterative blind deconvolution method and its applications', *Optics Letters*, Vol. 13, No. 7, July, pp. 547–549.

BATES, R.H.T. (1976), 'A stochastic image restoration procedure', *Optics Communications*, Vol. 19, No. 2, November, pp. 240–244.

BATES, R.H.T. (1982), 'Astronomical speckle imaging', *Physics Reports*, Vol. 90, pp. 203–297.

BATES, R.H.T. and CADY, F.M. (1980), 'Towards true imaging by wideband speckle interferometry', *Optics Communications*, Vol. 32, No. 3, March, pp. 365–369.

BATES, R.H.T. and DAVEY, B.L.K. (1987), 'Towards making shift-and-add a versatile imaging technique', In IDELL, P.S. (Ed.), *Digital Image Recovery and Synthesis*, Proceedings SPIE: Vol. 828, pp. 87–94.

BATES, R.H.T. and JIANG, H. (1991), 'Blind deconvolution – recovering the seemingly irrecoverable!', In GOODMAN, J.W. (Ed.), *International Trends In Optics*, Academic Press, San Diego, pp. 423–437.

BATES, R.H.T. and LANE, R.G. (1987), 'Deblurring should now be automatic', In *6th Pfefferkorn Conference on Image and Signal Processing in Electron Microscopy, Niagara Falls, Ontario, Canada*, April.

BATES, R.H.T. and McDONNELL, M.J. (1989), *Image Restoration and Reconstruction*, The Oxford Engineering Science Series; 16, Clarendon Press, Oxford. First published 1986. Corrected and updated 1989.

BATES, R.H.T. and MINARD, R.A. (1984), 'Compensation for multiple reflection', *IEEE Transactions on Sonics and Ultrasonics*, Vol. SU-31, No. 4, July, pp. 330–336.

BATES, J.H.T., FRIGHT, W.R. and BATES, R.H.T. (1984), 'Wiener filtering and cleaning in a general image processing context', *Monthly Notices of the Royal Astronomical Society*, Vol. 211, No. 1, November, pp. 1–14.

BATES, R.H.T., JIANG, H. and DAVEY, B.L.K. (1990a), 'Multidimensional system identification through blind deconvolution', *Multidimensional Systems and Signal Processing*, Vol. 1, pp. 127–142.

BATES, R.H.T., QUEK, B.K. and PARKER, C.R. (1990b), 'Some implications of zero sheets for blind deconvolution and phase retrieval', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 7, No. 3, March, pp. 468–479.

BENDAT, J.S. and PIERSOL, A.G. (1980), *Engineering applications of correlation and spectral analysis*, John Wiley & Sons, New York.

BIEMOND, J., LAGENDIJK, R.L. and MERSEREAU, R.M. (1990), 'Iterative methods for image deblurring', *Proceedings of the IEEE*, Vol. 78, No. 5, May, pp. 856–883.

BONES, P.J., PARKER, C.R., SATHERLEY, B.L. and WATSON, R.W. (1995), 'Deconvolution and phase retrieval with use of zero sheets', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 12. To appear in.

BOSE, N.K. (1990), 'Multidimensional signal processing', *Proceedings IEEE, special issue*, Vol. 78, No. 4.

BRACEWELL, R.N. (1978), *The Fourier Transform and its Applications*, McGraw-Hill, Tokyo.

BRIGHAM, E.O. (1974), *The Fast Fourier Transform*, Prentice-Hall.

BURCH, S.F., GULL, S.F. and SKILLING, J. (1983), 'Image restoration by a powerful maximum entropy method', *Computer Vision, Graphics, and Image Processing*, Vol. 23, No. 2, August, pp. 113–128.

CARRINGTON, W. (1990), 'Image restoration in 3-d microscopy with limited data', *Bioimaging and Two-Dimensional Spectroscopy*, Vol. 1205, pp. 64–71.

CASTLEMAN, K. (1979), *Digital Image Processing*, Prentice-Hall, New Jersey.

CONCHELLO, J. and HANSEN, E.W. (1989), 'Three-dimensional reconstruction of noisy confocal scanning microscope images', *New Methods in Microscopy and Low Light Imaging*, Vol. 1161, pp. 279–285.

CURTIS, S.R., OPPENHEIM, A.V. and LIM, J.S. (1985), 'Signal reconstruction from Fourier transform sign information', *IEEE Transactions Acoustics, Speech, Signal Processing*, Vol. ASSP-33, No. 3, June, pp. 643–657.

DAVEY, B.L.K. (1989), *Advances In Blind Deconvolution*, PhD thesis, University of Canterbury, Christchurch, New Zealand.

DAVEY, B.L.K., LANE, R.G. and BATES, R.H.T. (1989), 'Blind deconvolution of noisy complex-valued image', *Optics Communications*, Vol. 69, No. 5–6, 15 Jan., pp. 353–356.

DEMPSTER, A.P., LAIRD, N.M. and RUBIN, D.B. (1977), 'Maximum likelihood from incomplete data via the EM algorithm', *J. R. Stat. Soc. B*, Vol. 39, pp. 1–37.

FIENUP, J.R. (1982), 'Phase retrieval algorithms: a comparison', *Applied Optics*, Vol. 21, No. 15, August, pp. 2758–2769.

FISH, D.A., BRINICOMBE, A.B. and PIKE, E.R. (1995), 'Blind deconvolution by means of the Richardson-Lucy algorithm', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 12, No. 1, January, pp. 58–65.

FISHER, R.A. (1922), 'On the mathematical foundations of theoretical statistics', *Philos Trans Royal Soc London*, Vol. Ser A, pp. 222:309–368.

FLETCHER, R. (1987), *Practical Methods of Optimization*, Wiley, New York.

GERCHBERG, R.W. and SAXTON, W.O. (1972), 'A practical algorithm for the determination of phase from image and diffraction plane pictures', *Optik*, Vol. 35, No. 2, April, pp. 237–246.

GHIGLIA, D.C., ROMERO, L.A. and MASTIN, G.A. (1993), 'Systematic approach to two-dimensional blind deconvolution by zero-sheet separation', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 10, No. 5, May, pp. 1024–1036.

GILL, P.E. and W. MURRAY, E. (1974), *Numerical Methods for constrained Optimisation*, Academic, London.

GONZALEZ, R.C. and WOODS, R.E. (1992), *Digital Image Processing*, Addison-Wesley.

GOODMAN, J.W. (1968), *Fourier Optics*, McGraw-Hill, New York.

GULL, S.F. and SKILLING, J. (1984), 'Maximum entropy method in image processing', *Proceedings IEE Part F*, Vol. 131, No. 6, October, pp. 645 – 659.

HALL, E.L. (1979), *Computer Image Processing and Recognition*, Academic Press, New York.

HAYKIN, S. (1983), *Communication Systems*, John Wiley & Sons, New York. Second edition.

HELSTROM, C.W. (1967), 'Image restoration by the method of least squares', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 57, pp. 297–303.

HOLMES, T.J. (1992), 'Blind deconvolution of quantum-limited incoherent imagery: maximum-likelihood approach', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 9, No. 7, July, pp. 1052–1061.

HOLMES, T.J. and LIU, Y.H. (1989), 'Richardson-Lucy/maximum-likelihood image restorations for fluorescence microscopy: further testing', *Applied Optics*, Vol. 28, pp. 4930–4938.

IZRAELEVITZ, D. and LIM, J.S. (1987), 'A new direct algorithm for image reconstruction from Fourier transform magnitude', *IEEE Transactions Acoustics, Speech, Signal Processing*, Vol. ASSP-35, No. 4, April, pp. 511–519.

JIANG, H. and BONES, P.J. (1991), 'Estimation of best support size in iterative blind deconvolution', In *"6th New Zealand Image Processing Workshop*, Lower Hutt, August, pp. 127–132.

KATSAGGELOS, A.K. (1991), *Digital Image Restoration*, Springer Series in information Sciences, Springer-Verlag, Berlin.

LABEYRIE, A. (1970), 'Attainment of diffraction-limited resolution in large telescopes by Fourier analysing speckle patterns in star images', *Astronomy and Astrophysics*, Vol. 6, No. 1, May, pp. 85–87.

LAGENDIJK, R.L., KATSAGGELOS, A.K. and BIEMOND, J. (1988), 'Iterative identification and restoration', In *IEEE International Conference ASSP*, pp. 992–995.

LAGENDIJK, R.L., BIEMOND, J. and BOEKEE, D.E. (1990), 'Identification and restoration of noisy blurred images using the expectation-maximization algorithm', *IEEE Transactions Acoustics, Speech Signal Process.*, Vol. 38, pp. 1180–1191.

LANE, R. (1988), *Blind Deconvolution and Phase Retrieval*, PhD thesis, University of Canterbury, Christchurch, New Zealand.

LANE, R.G. (1992), 'Blind deconvolution of speckle images', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 9, No. 9, September, pp. 1058–1514.

LANE, R.G. and BATES, R.H.T. (1987a), 'Automatic multidimensional deconvolution', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 4, No. 1, January, pp. 180–188.

LANE, R.G. and BATES, R.H.T. (1987b), 'Relevance for blind deconvolution of recovering Fourier magnitude from phase', *Optics Communications*, Vol. 63, No. 1, July, pp. 11–14.

LANE, R.G., FRIGHT, W.R. and BATES, R.H.T. (1987), 'Direct phase retrieval', *IEEE Transactions Acoustics, Speech, Signal Processing*, Vol. ASSP-35, No. 4, April, pp. 520–526.

LANGE, K., BAHN, M. and LITTLE, R. (1987), 'A theorerical study of some maximum likelihood algorithms for emission and transmission tomography', *IEEE Trans. Med. Imag.*, Vol. MI-6, pp. 106–114.

LEVIN, B.J. (1964), *Distribution of Zeros of Entire Functions*, American Mathematical Society, Providence, Rhode Island.

LEWITT, R.M. and MUEHLLEHNER, G. (1986), 'Accelerated iterative reconstruction for positron emission tomography based on the EM algorithm for maximum likelihood estimation', *IEEE Trans. Med. Imag.*, Vol. MI-5, pp. 16–22.

LUCY, L.B. (1974), 'An iterative technique for the rectification of observed distributions', *Astron. J.*, Vol. 79, pp. 745–754.

McCALLUM, B.C. (1990a), *Advances in Phase Retrieval and Blind Deconvolution*, PhD thesis, University of Canterbury, Christchurch, New Zealand.

McCALLUM, B.C. (1990b), 'Blind deconvolution by simulated annealing', *Optics Communications*, Vol. 75, No. 2, 15 Feb., pp. 101–105.

McCALLUM, B.C. and RODENBURG, J.M. (1993), 'Simultaneous reconstruction of object and aperture functions from multiple far-field intensity measurements', *Journal of the Optical Society of America A: Optics and Image Science*, No. 10, pp. 231–239.

MENDEL, J.M. (1985), 'How to include prespecified horizons into minimum-variance deconvolution and maximum-likelihood deconvolution', *Geophysics*, Vol. 50, pp. 1510–1512.

MENDEL, J.M. (1990), *Maximum-Likelihood Deconvolution*, Springer-Verlag, New York.

METROPOLIS, N., ROSENBLUTH, A., ROSENBLUTH, M., TELLER, A. and TELLER, E. (1953), 'Equation of state calculations by fast computing machines', *Journal of Chemical Physics*, Vol. 21, No. 4, June, pp. 1087–1092.

MILLANE, R.P. (1990a), 'Phase retrieval in crystallography and optics', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 7, No. 3, March, pp. 394–411.

MILLANE, R.P. (1990b), 'Redundancy in multidimensional deconvolution and phase retrieval', In *Digital image synthesis and inverse optics*, "Proceedings of the SPIE, San Diego, July, pp. 227–236.

MILLANE, R.P., BONES, P.J. and JIANG, H. (1994), 'Blind deconvolution for multidimensional images', In *IEEE International Conference ASSP*, Adelaide, Australia, April, pp. 445–448.

MISELL, D.L. (1978), 'The phase problem in electron microscopy', In COSSLETT, V.E. and BARER, R. (Eds.), *Advances in optical and electron microscopy, Volume 7*, Academic Press, London, pp. 185–279.

NAKAJIMA, N. (1993), 'Blind deconvolution using the maximum likelihood estimation and the iterative algorithm', *Optics Communications*, Vol. 100, 1 July, pp. 59–66.

NASHED, M.Z. (1981), *Ill-posed Problems: Theory and Practice*, Reidel, Dordrecht.

NAVARRO, R., FUENTES, F.J. and NIETO-VESPERINAS, M. (1989), 'Simulated annealing image reconstruction in photon-limited stellar speckle interferometry', *Astronomy and Astrophysics*, Vol. 208, No. 1/2, January, pp. 374–380.

NIETO-VESPERINAS, M. and BENDEZ, J.A. (1986), 'Phase retrieval by monte-carlo methods', *Optics Communications*, Vol. 59, No. 4, September(15), pp. 249–254.

NIETO-VESPERINAS, M., NAVARRO, R. and FUENTES, F.J. (1988), 'Performance of a simulated-annealing algorithm for phase retrieval', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 5, No. 1, January, pp. 30–38.

OPPENHEIM, A.V., WILLSKY, A.S. and YOUNG, I.T. (1983), *Signals and Systems*, Prentice-Hall, New Jersey.

PALEY, R.E.A.C. and WIENER, N. (1934), *Fourier transforms in the complex domain*, American Mathematical Society, Providence, R. I.

PARKER, C.R. (1994), *Zero-based Phase Retrieval*, PhD thesis, University of Canterbury, Christchurch, New Zealand.

PARKER, C.R. and BONES, P.J. (1992a), 'Image deblurring via zero sheets', In *Seventh New Zealand Image Processing Workshop*, University of Canterbury, Christchurch, August, pp. 5–10.

PARKER, C.R. and BONES, P.J. (1992b), 'Convergence of iterative phase retrieval improved by utilizing zero sheets', *Optics Communications*, Vol. 92, No. 4,5,6, September, pp. 209–214.

POLAK, E. (1971), *Computational Methods in Optimization*, Academic Press, New York.

PRATT, W.K. (1991), *Digital image Processing*, John Wiley & Sons, New York.

PRESS, W.H., FLANNERY, B.P., TEUKOLSKY, S.A. and VETTERLING, W.T. (1992), *Numerical Recipes in C*, Cambridge University Press, Cambridge.

RAMACHANDRAN, G.N. and SRINIVASAN, R. (1970), *Fourier Methods in Crystallography*, Wiley-Interscience, New York.

REQUICHA, A.A.G. (1980), 'The zeros of entire functions: Theory and engineering applications', *Proceedings of the IEEE*, Vol. 68, No. 3, March, pp. 308–328.

RIAD, S.M. (1986), 'The deconvolution problem: an overview', *Proceedings of the IEEE*, Vol. 74, No. 1, pp. 82–85.

RICHARDSON, W.H. (1972), 'Bayesian-based iterative method of image restoration', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 62, pp. 55–59.

RODDIER, F. (1988), 'Interferometric imaging in optical astronomy', *Physics Reports*, Vol. 170, No. 2, November, pp. 97–166.

ROSENFELD, A. and KAK, A.C. (1982), *Digital picture processing*, Academic press, New York.

ROYSAM, B., SHRAUNER, J.A. and MILLER, M.I. (1988), 'Bayesian imaging using good's roughness measure implementation on a massively parallel processor', In *IEEE International Conference ASSP*, New York, pp. 932–935.

SANZ, J.L.C. and HUANG, T.S. (1985), 'Polynomial system of equations and its applications to the study of the effect of noise on multidimensional Fourier transform phase retrieval from magnitude', *IEEE Transactions Acoustics, Speech, Signal Processing*, Vol. ASSP-33, No. 4, August, pp. 997–1004.

SATHERLEY, B.L. (1994), *Zero-based ensemble deconvolution and EEG spectral topography*, PhD thesis, University of Canterbury, Christchurch, New Zealand.

SATHERLEY, B.L. and BONES, P.J. (1994), 'Zero tracks for blind deconvolution of blurred ensembles', *Applied Optics*, Vol. 33, No. 11, pp. 2197–2205.

SATHERLEY, B.L. and PARKER, C.R. (1993), 'Two-dimensional image reconstruction from zero sheets', *Optics Letters*, Vol. 18, No. 23, December, pp. 2053–2055.

SELDIN, J.H. and FIENUP, J.R. (1990), 'Iterative blind deconvolution algorithm applied to phase retrieval', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 7, No. 3, March, pp. 428–433.

SHEPP, L.A. and VARDI, Y. (1982), 'Maximum likelihood reconstruction for emission tomography', *IEEE Trans. Med. Imag.*, Vol. MI-1, pp. 113–122.

STOCKHAM, JR., T.G., CANNON, T.M. and INGEBRETSON, R.B. (1975), 'Blind deconvolution through digital signal processing', *Proceedings of the IEEE*, Vol. 63, No. 4, April, pp. 678–692.

SYNDER, D.L. (1975), *Random Point Processes*, Wiley, New York.

SYNDER, D.L. and MILLER, M.I. (1985), 'The use of sieves to stabilize images produced with the EM algorithm for emission tomography', *IEEE Trans. Nucl. Sci.*, Vol. NS-32, pp. 3864–3872.

TEKALP, A.M. and KAUFMAN, H. (1988), 'On statistical identification of a class of linear space-invariant image blurs using nonminimum-phase ARMA models', *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 36, pp. 1360–1363.

TEKALP, A.M., KAUFMAN, H. and WOODS, J.W. (1986), 'Identification of image and blur parameters for the restoration of noncausal blurs', *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 34, pp. 963–972.

THORPE, C.W. (1990), *Analysis of speech and other sounds*, PhD thesis, University of Canterbury, Christchurch, New Zealand.

TIKHONOV, A.N. and ARSENIN, V.Y. (1977), *Solutions of Ill-posed Problems*, Wiley, New York.

WATSON, R.W., PARKER, C.R. and BONES, P.J. (1992), 'Demonstration of two-dimensional consistent deconvolution using zeros', *Optics Communications*, Vol. 93, No. 5,6, October, pp. 359–365.

YANG, Y., GALATSANOS, N.P. and STARK, H. (1994), 'Projection-based blind deconvolution', *Journal of the Optical Society of America A: Optics and Image Science*, Vol. 11, No. 9, September, pp. 2401–2409.